EECE 5639 Computer Vision I

Lecture 13

Planar Unwarping:

- **2D-2D Homography,**
- **Feature Matching,**
- **Homography Estimation**

Project 2 is out …

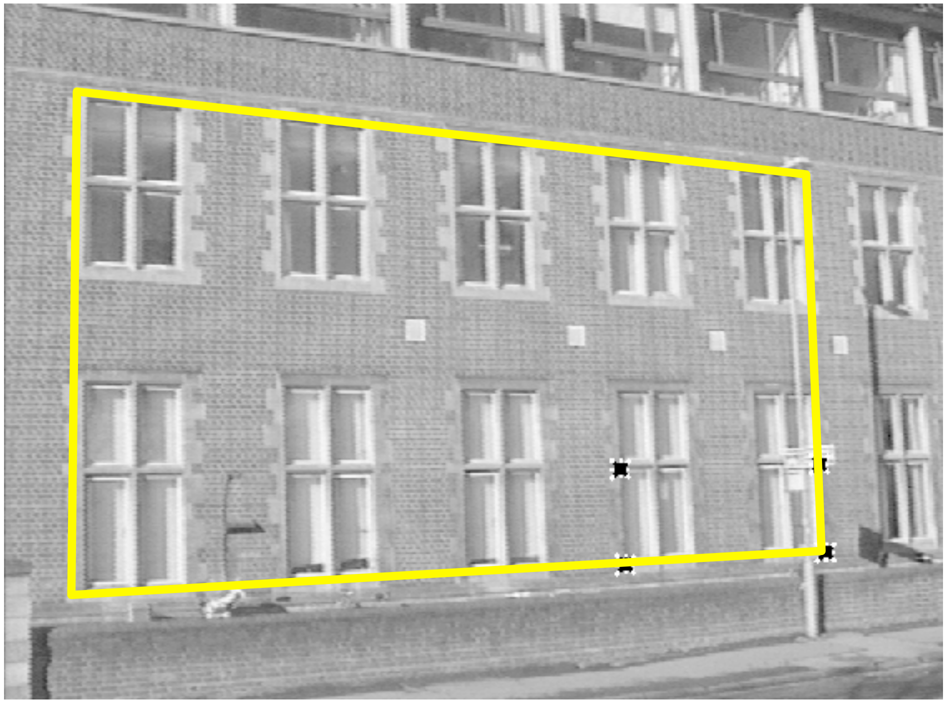
Next Class

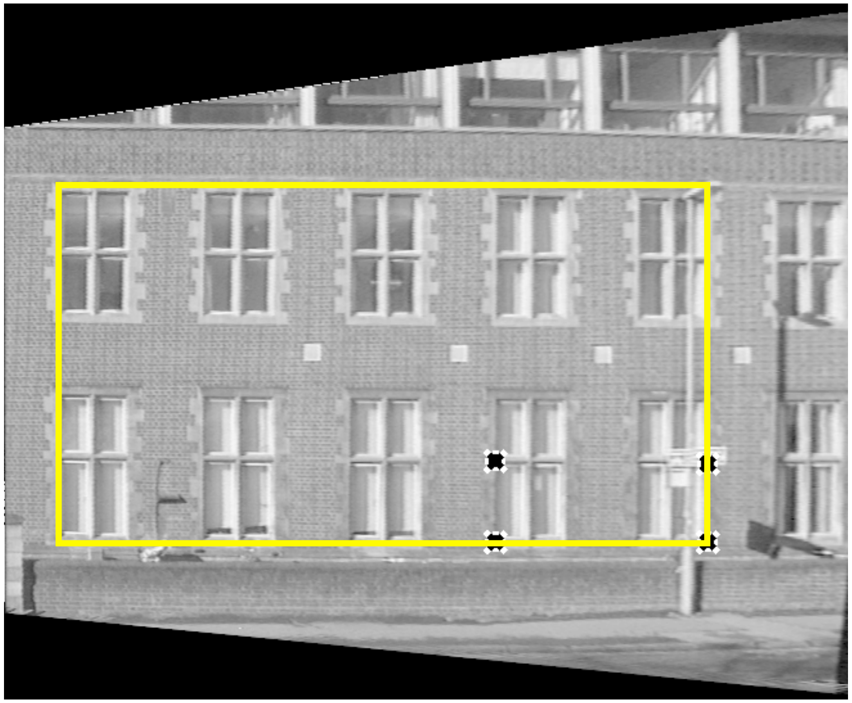
- **Unwarping using Matlab**
- **Stereo**

Planar Unwarping

Thanks: Several of these slides are from Bob Collins

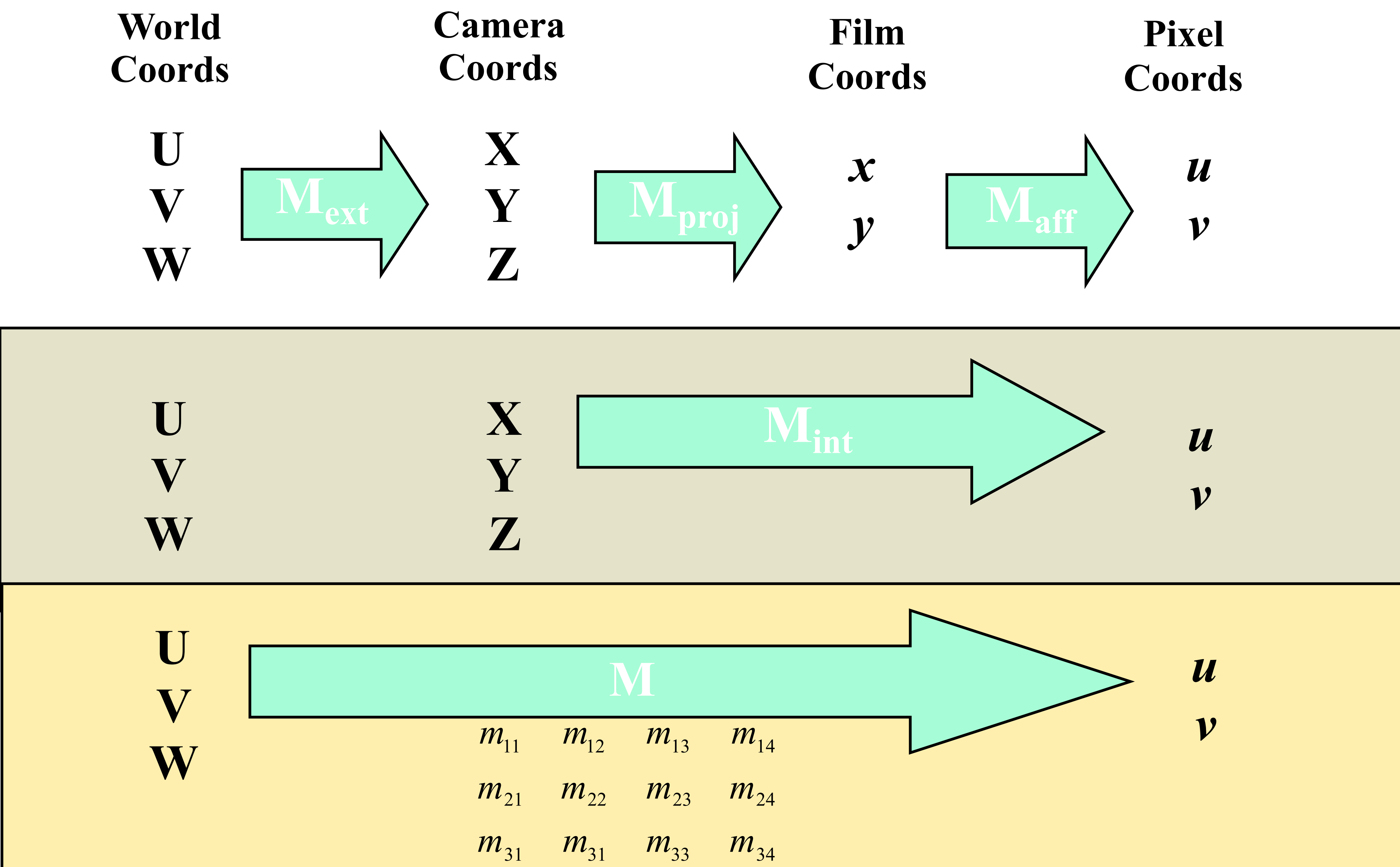
Remove Perspective Distortion





from Hartley & Zisserman

Review : Forward Projection

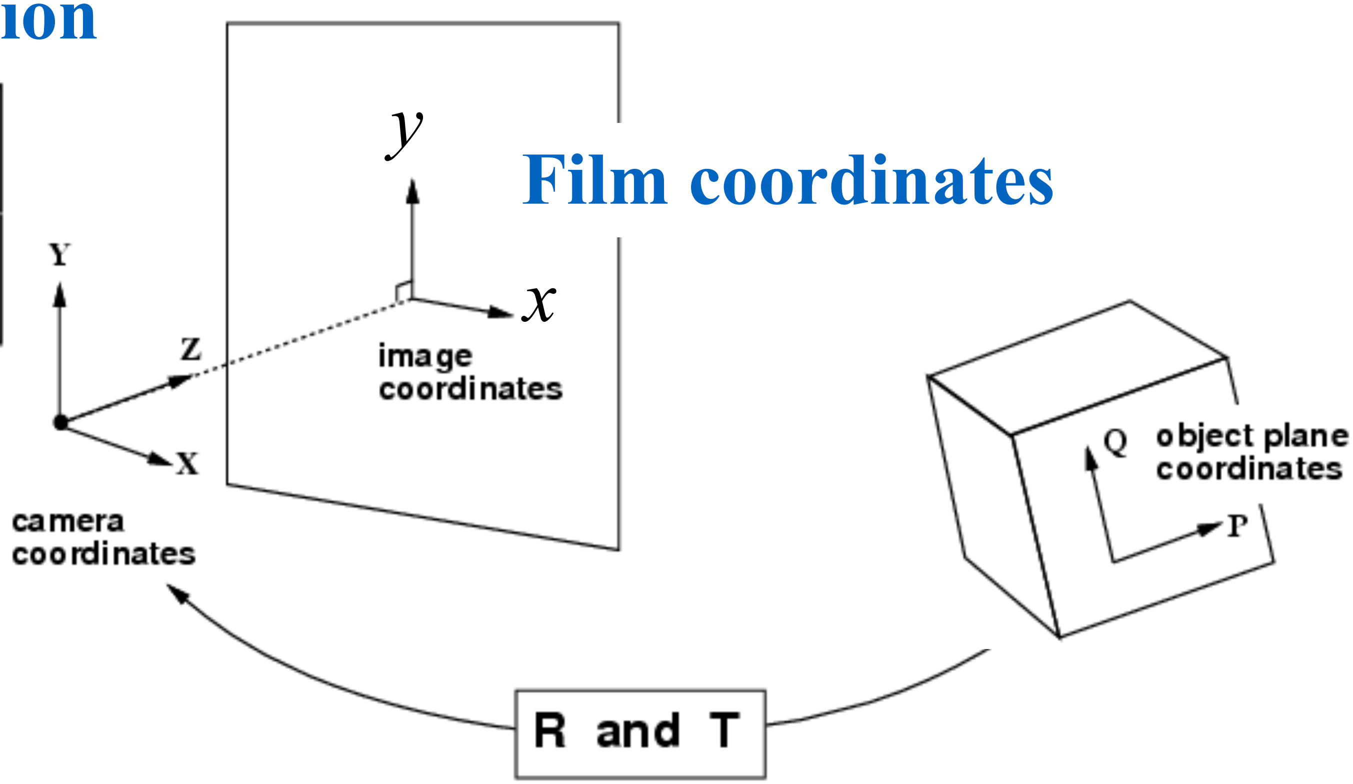


Projection of Points on Planar Surface

Perspective projection

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Film coordinates



$$\begin{bmatrix} p \\ q \\ 0 \\ 1 \end{bmatrix}$$

Point on plane

Rotation + Translation

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projection of Planar Points

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \boxed{r_{13}} & t_x \\ r_{21} & r_{22} & \boxed{r_{23}} & t_y \\ r_{31} & r_{32} & \boxed{r_{33}} & t_z \\ 0 & 0 & \boxed{0} & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ \boxed{0} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & \boxed{0} \\ 0 & f & 0 & \boxed{0} \\ 0 & 0 & 1 & \boxed{0} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \\ \boxed{0} & \boxed{0} & \boxed{1} \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

Projection of Planar Points (cont)

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} fr_{11} & fr_{12} & ft_x \\ fr_{21} & fr_{22} & ft_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

Homography H (planar projective transformation)

Special Case : Frontal Plane

What if the planar surface is perpendicular to the optic axis (Z axis of camera coord system)?

Then world rotation matrix simplifies:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Frontal Plane

So the homography for a frontal plane simplifies:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} fr_{11} & fr_{12} & ft_x \\ fr_{21} & fr_{22} & ft_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

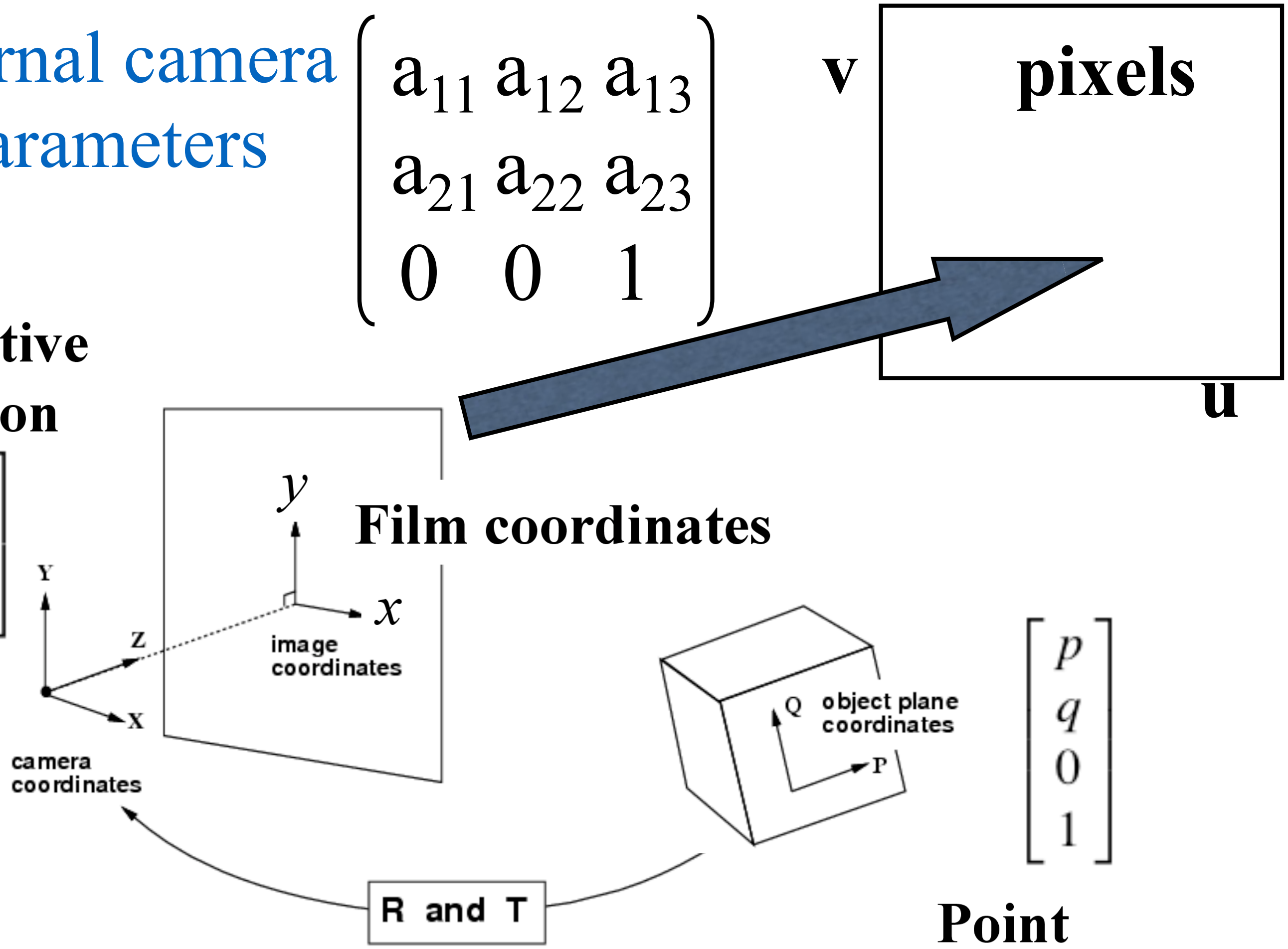
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f\cos\theta & -f\sin\theta & ft_x \\ f\sin\theta & f\cos\theta & ft_y \\ 0 & 0 & t_z \end{bmatrix} \begin{bmatrix} p \\ q \\ 1 \end{bmatrix}$$

Similarity Transformation!

Convert to Pixel Coords

Internal camera parameters

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix}$$



Perspective projection

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Point on plane

$$\begin{bmatrix} p \\ q \\ 0 \\ 1 \end{bmatrix}$$

Rotation + Translation

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Recall: Transformation Groups

A mathematical **group** G is composed of a set of elements and an associative operator * such that:

1) The set is closed under operator *

$$A \in G \text{ and } B \in G \rightarrow A * B \in G$$

2) There exists an identity element I such that

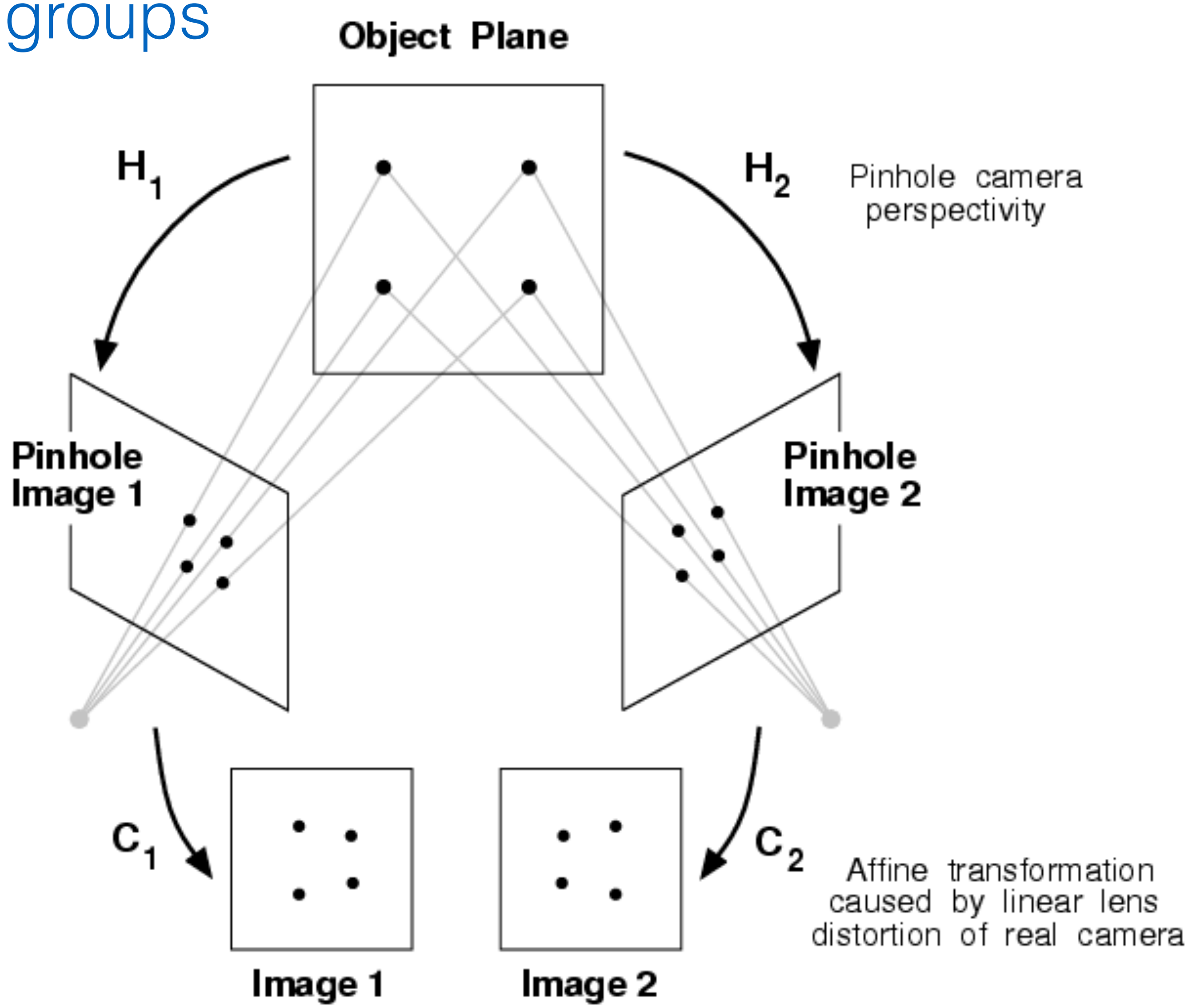
$$A * I = I * A = A$$

3) Each element A has an inverse A^{-1} such that

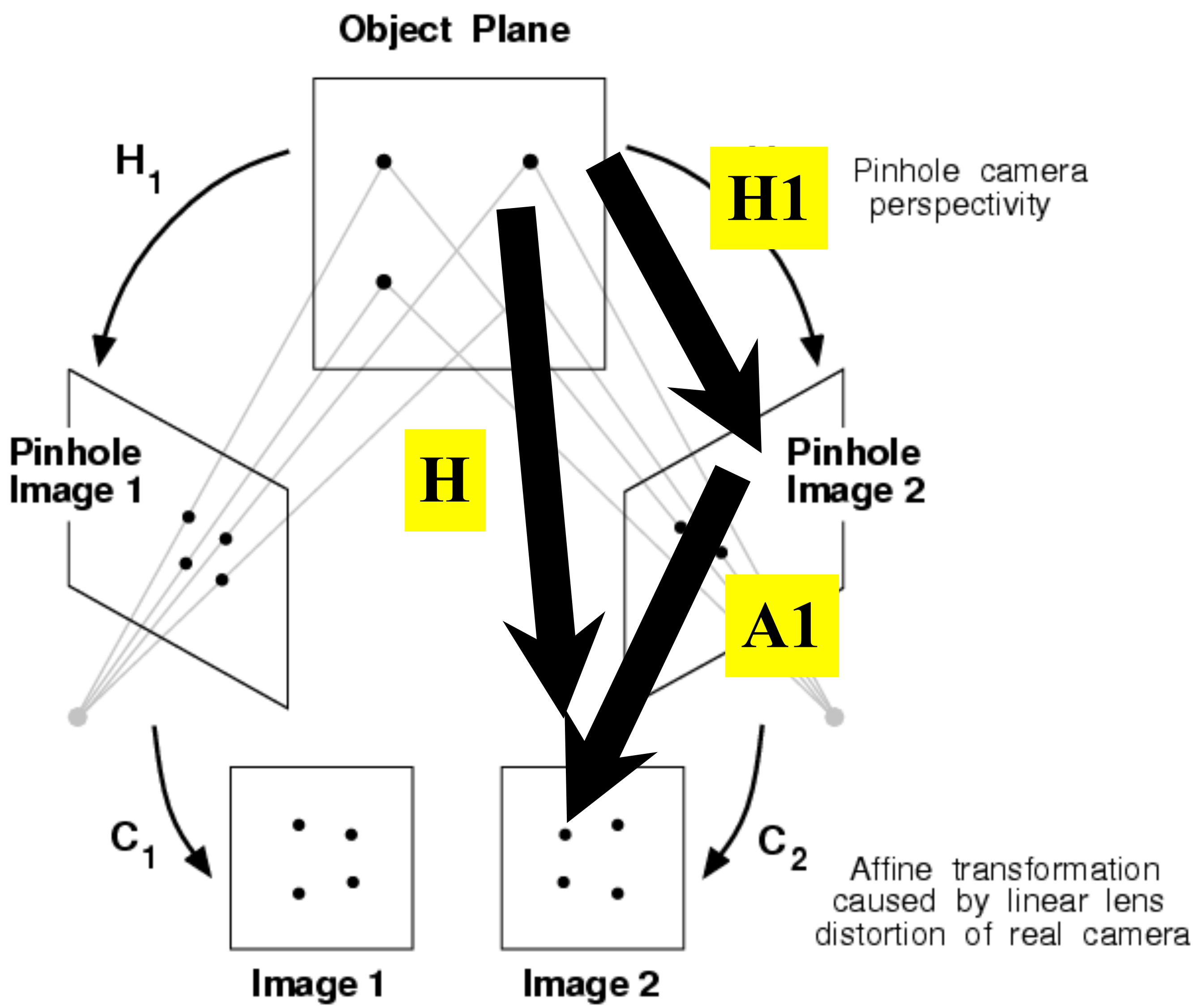
$$A^{-1} * A = A * A^{-1} = I$$

Planar Projection Diagram

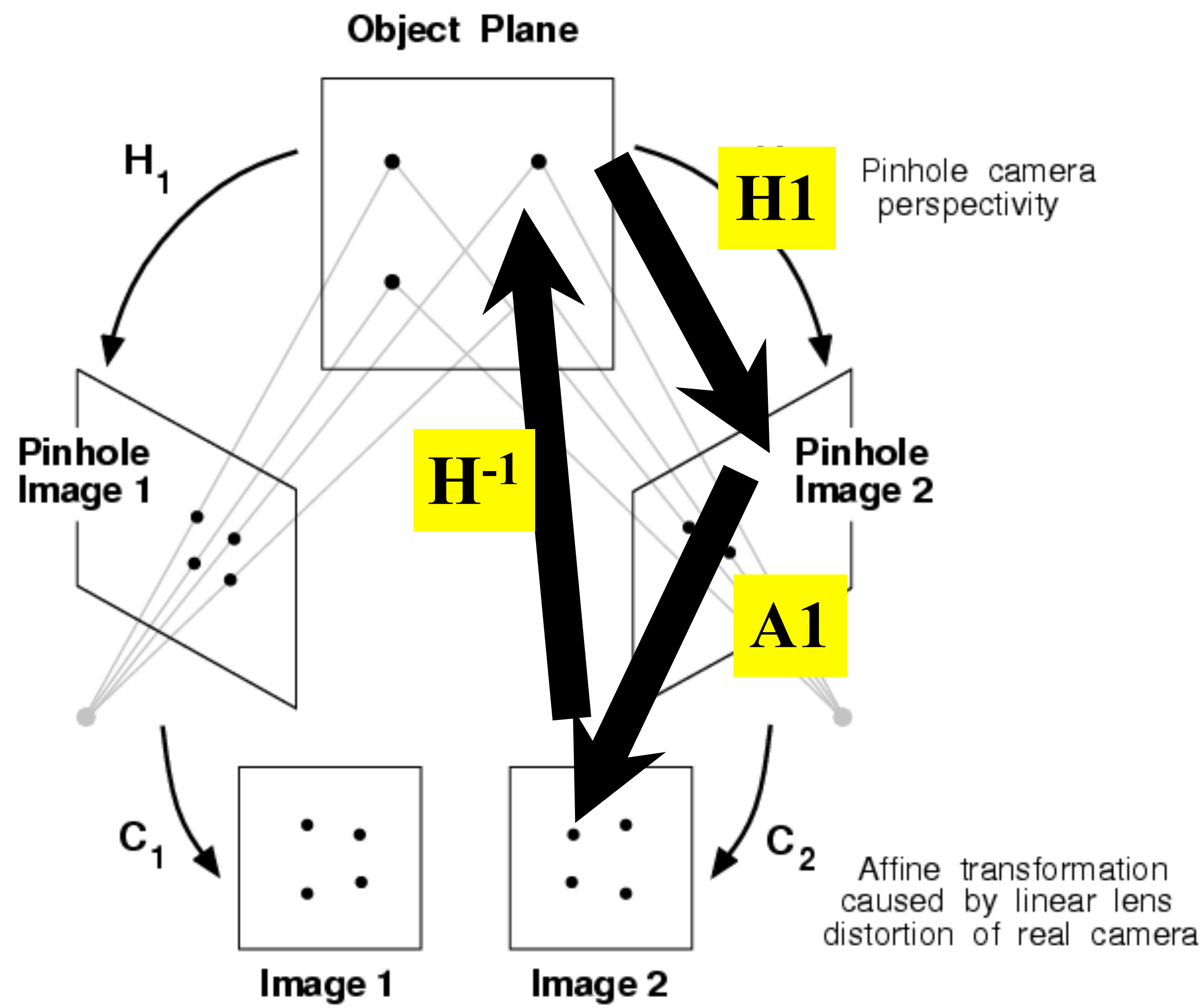
Here's where transformation groups get useful!



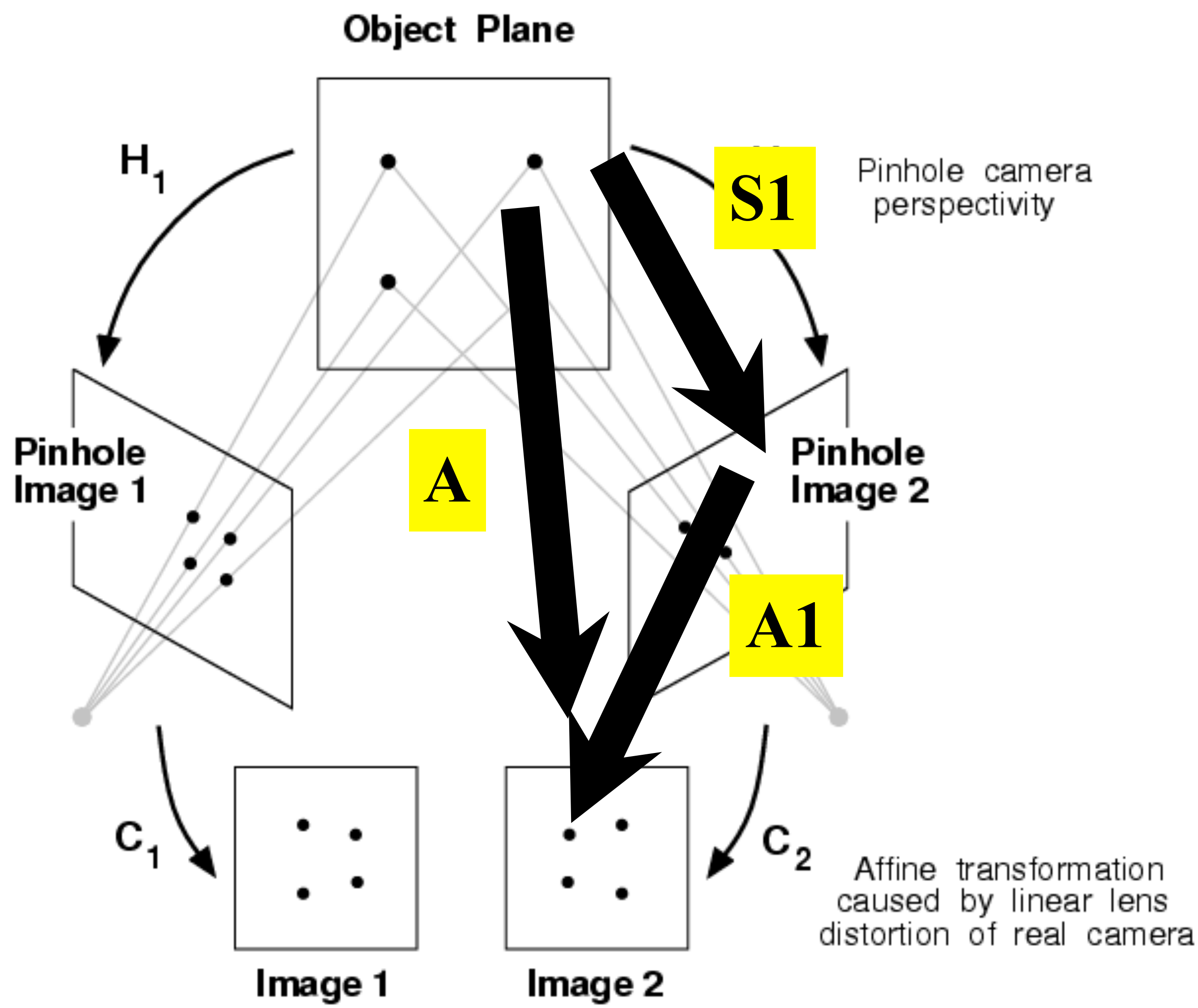
General Planar Projection



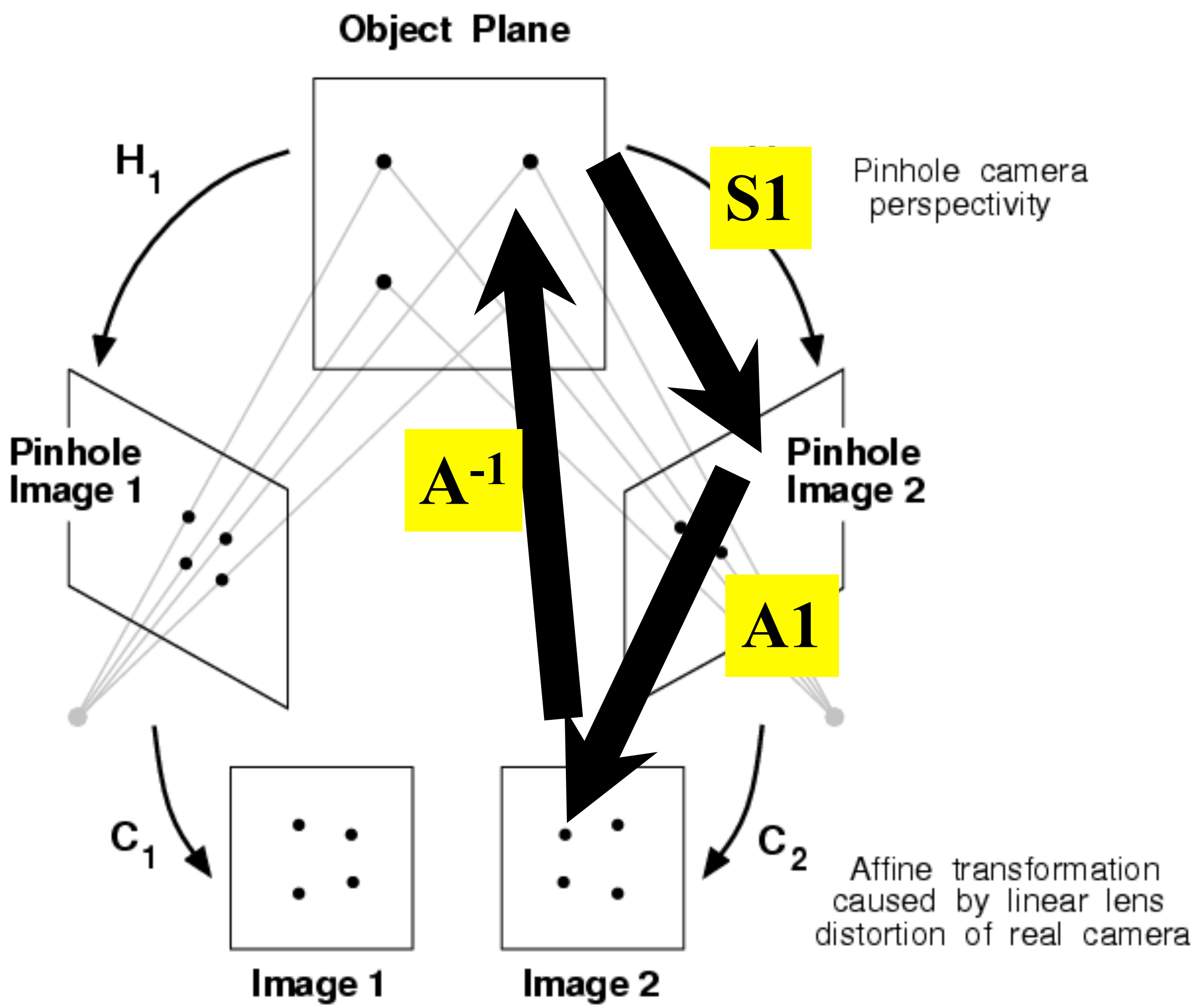
General Planar Projection



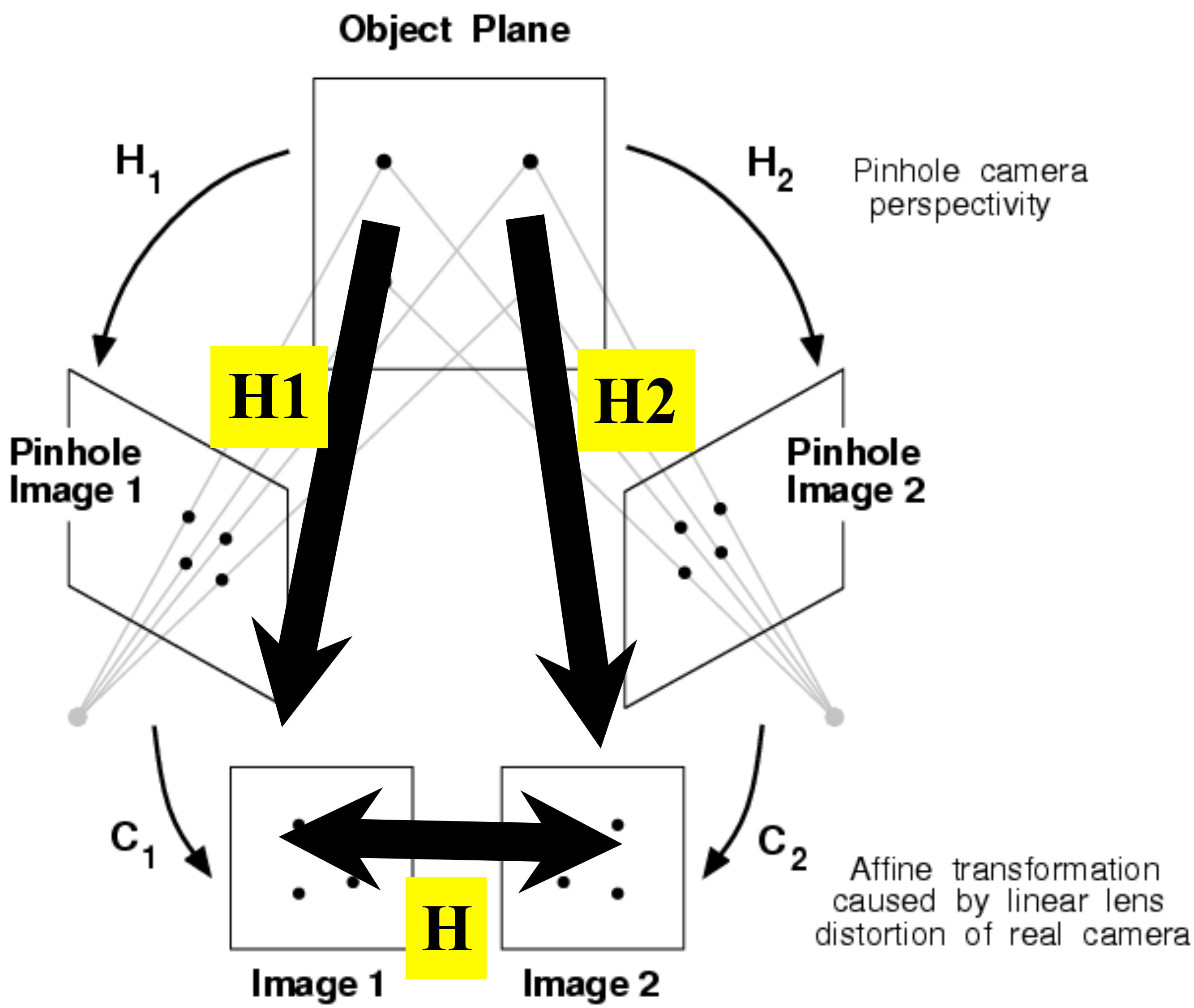
Frontal Plane Projection



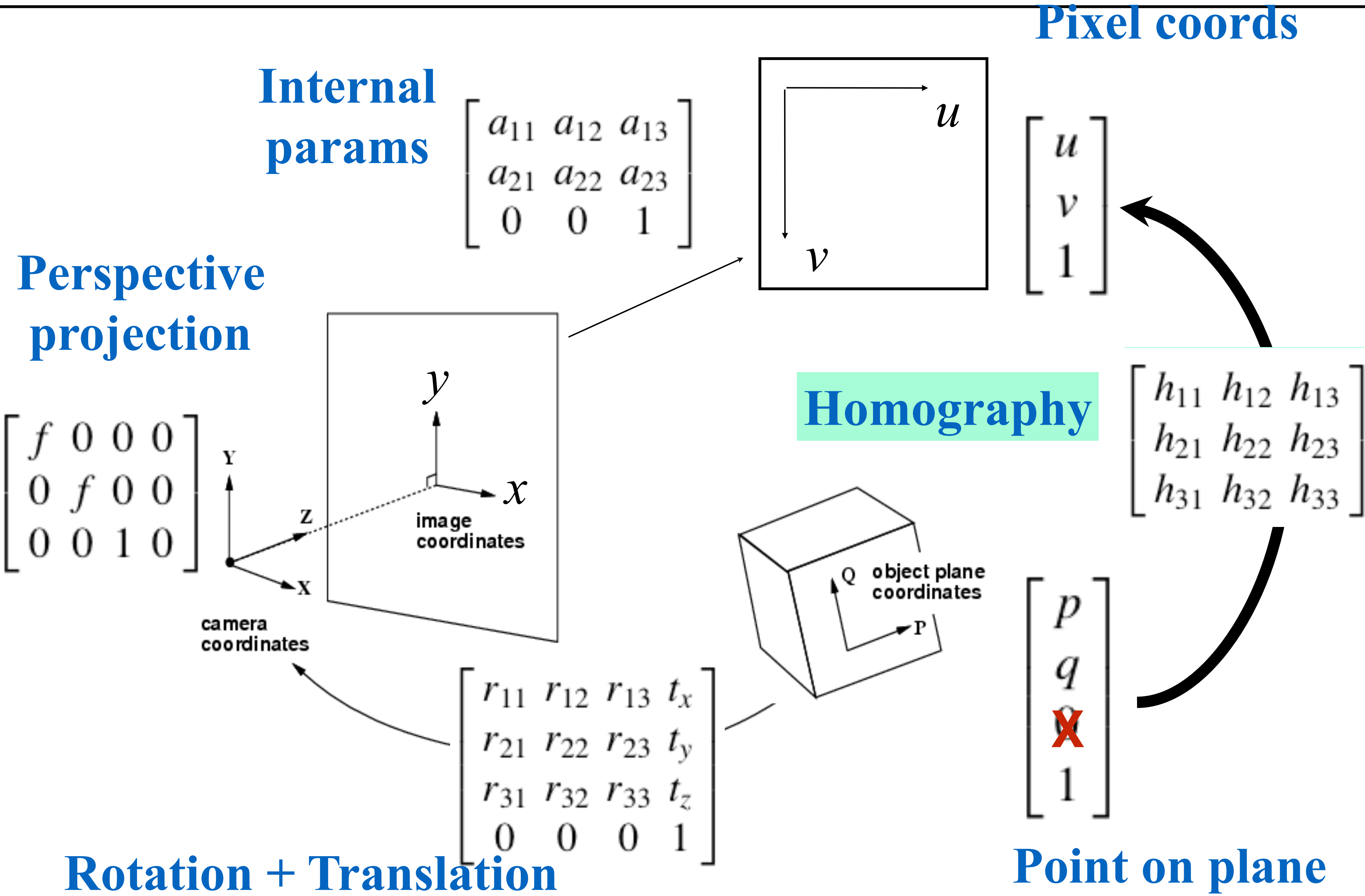
Frontal Plane Projection



General Planar Projection



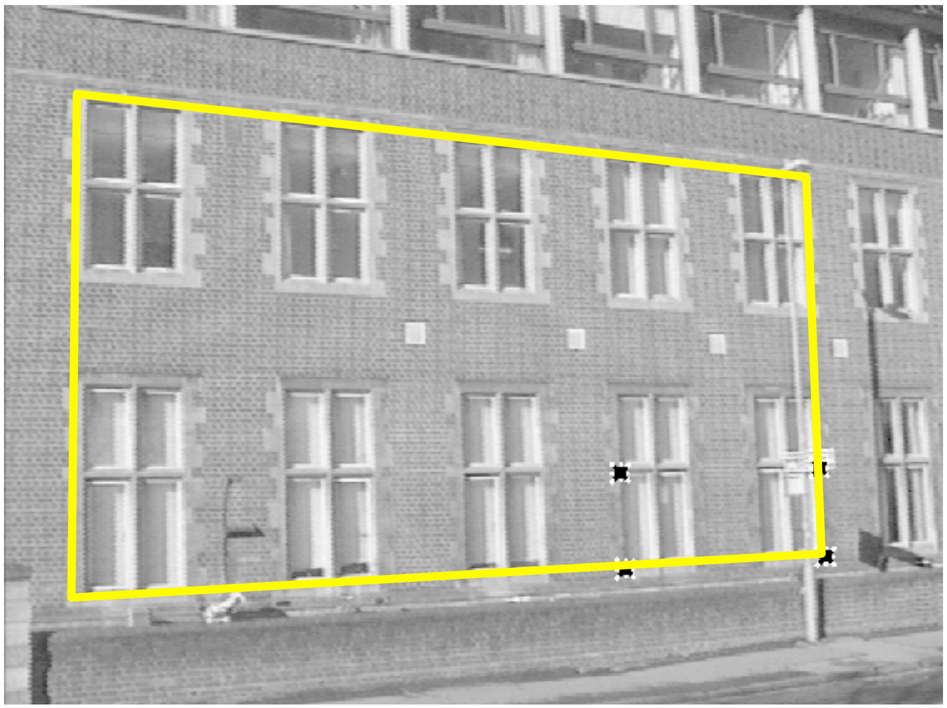
Summary: Planar Projection

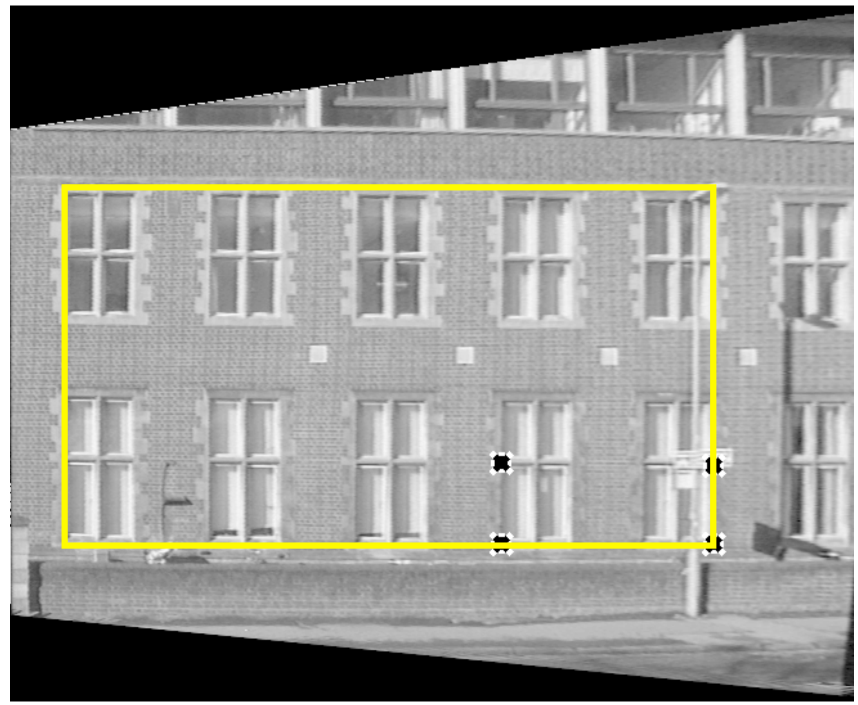


Recall Theorem:

A projective transformation of Pn onto itself is completely determined by its action on n+2 points.

Applying Homographies:Remove Perspective Distortion

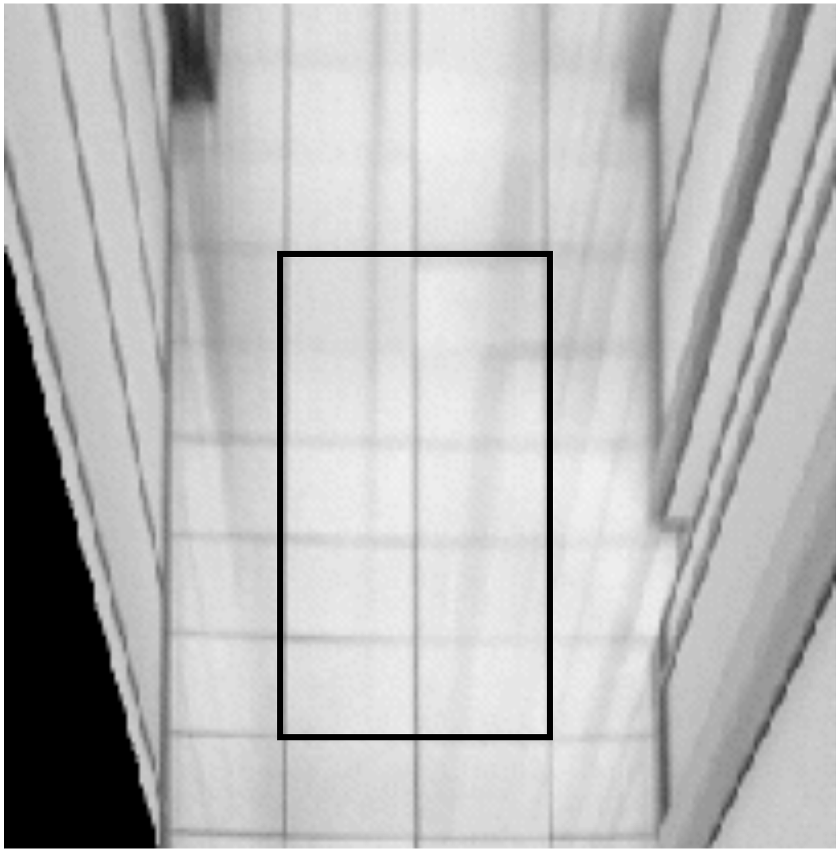




from Hartley & Zisserman

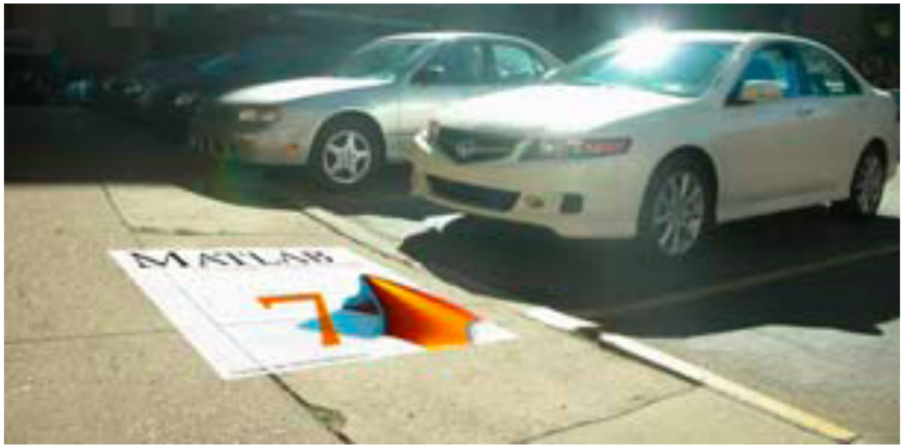
4 point correspondences suffice for the planar building facade

Homographies for Bird's-eye Views



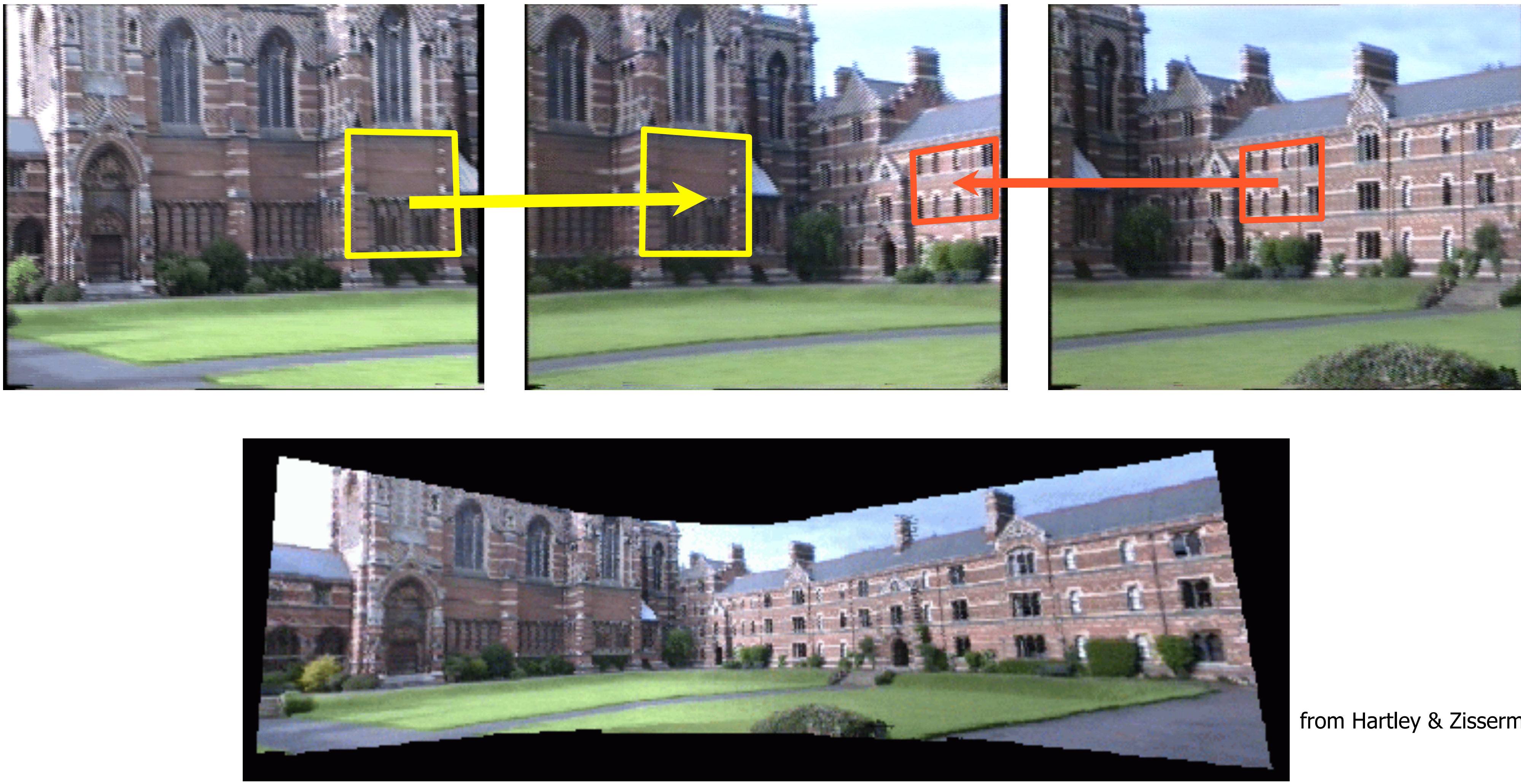
from Hartley & Zisserman

Homographies for placing on an image



Ekapol Chuangsuwanich, CMU (Slide from K. Grauman, UTA)

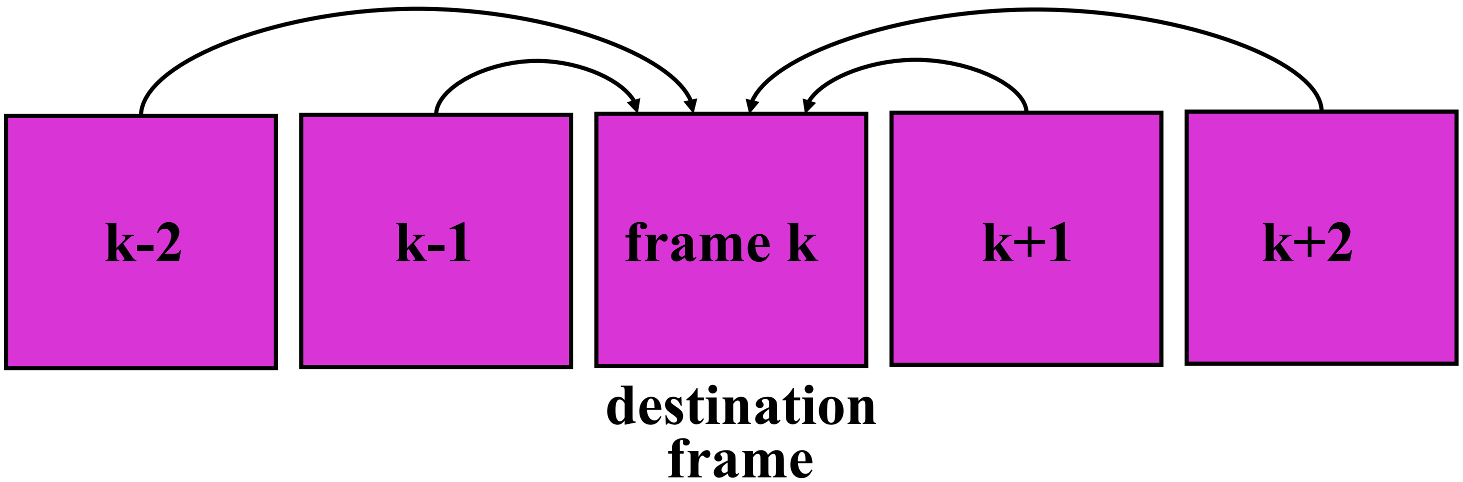
Homographies for Mosaicing



from Hartley & Zisserman

Applications: Stabilization

Given a sequence of video frames, warp them into a common image coordinate system.

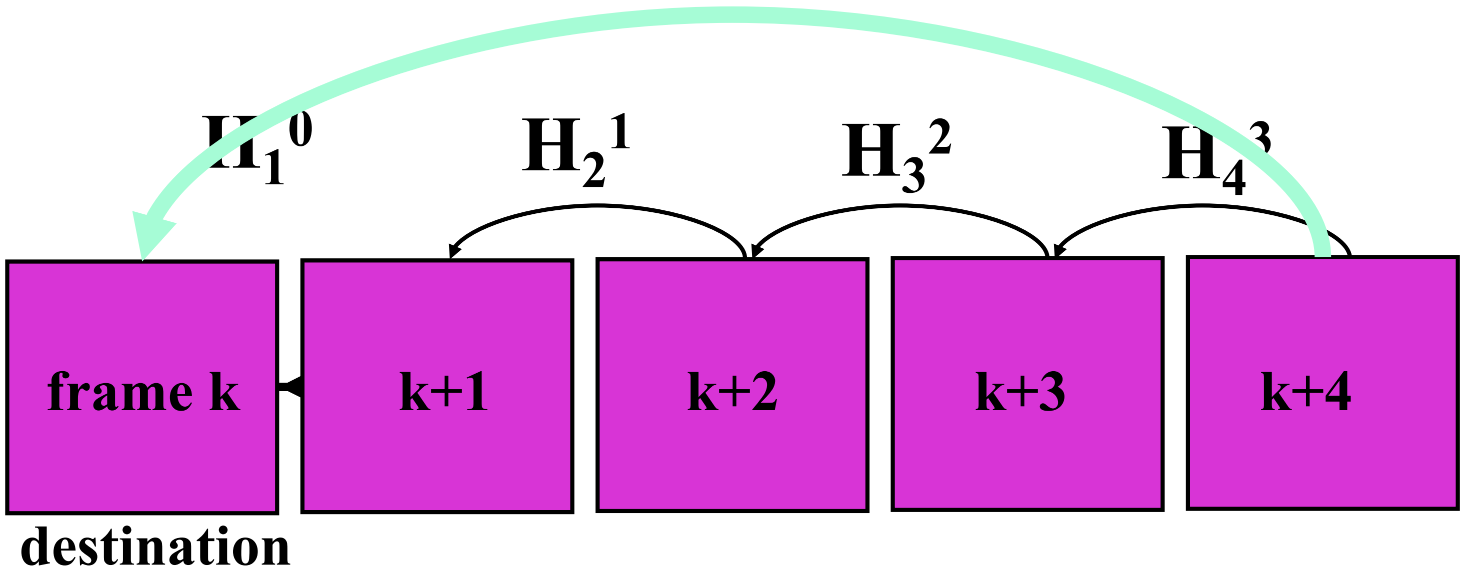


This “stabilizes” the video to appear as if the camera is not moving.

Stabilization by Chaining

What if the reference image does not overlap with all the source images? As long as there are pairwise overlaps, we can chain (compose) pairwise homographies.

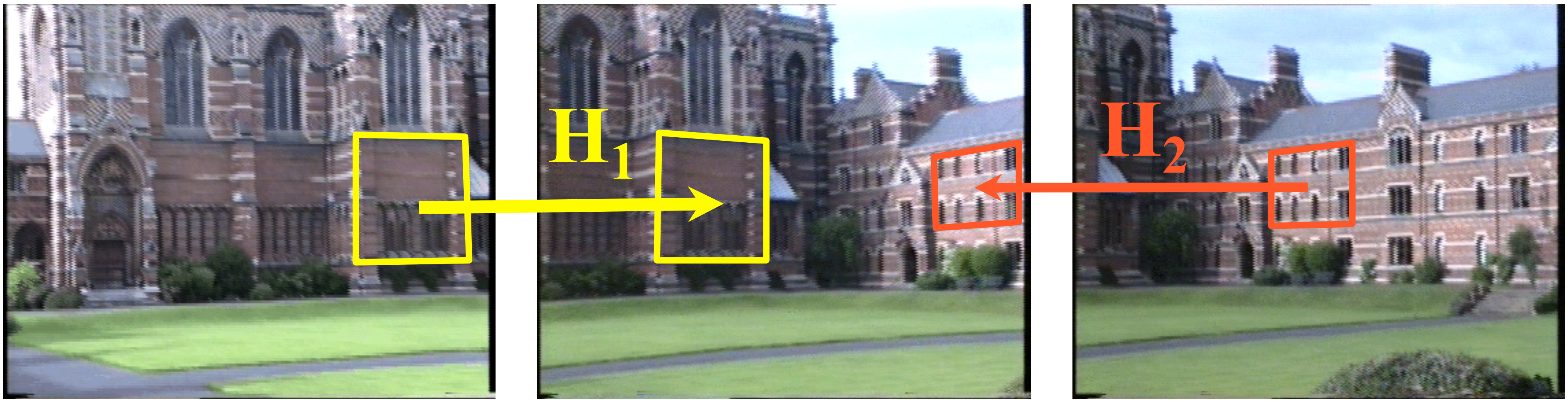
$$H_4^0 = H_1^0 * H_2^1 * H_3^2 * H_4^3$$



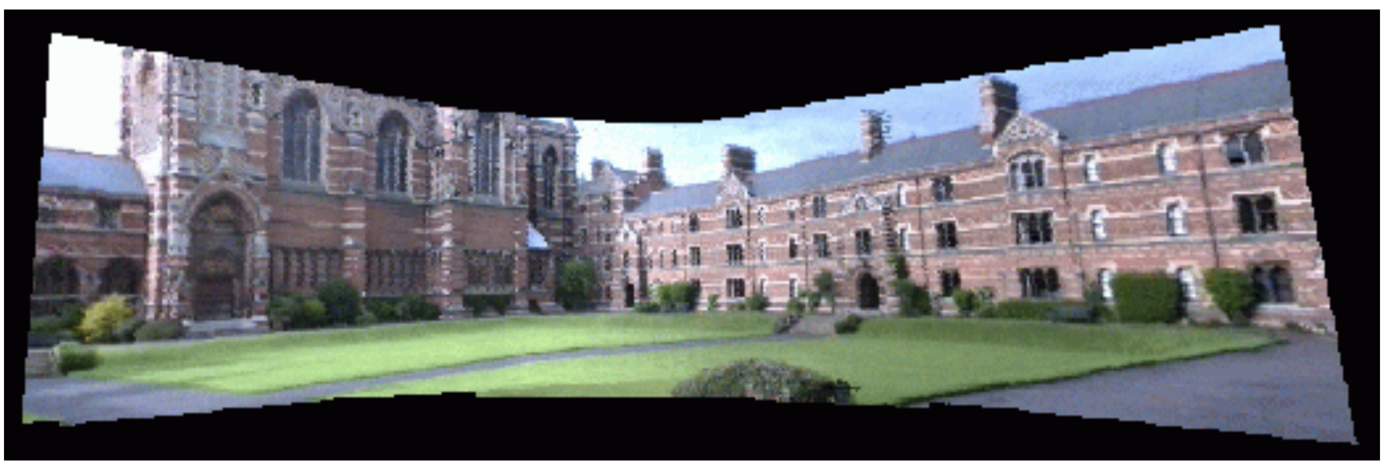
Not recommended for long sequences, as alignment errors accumulate over time.

Applications: Mosaicing

Source 1 **Destination image** **Source 2**



from Hartley & Zisserman



Mosaic

Note on Planar Mosaicing

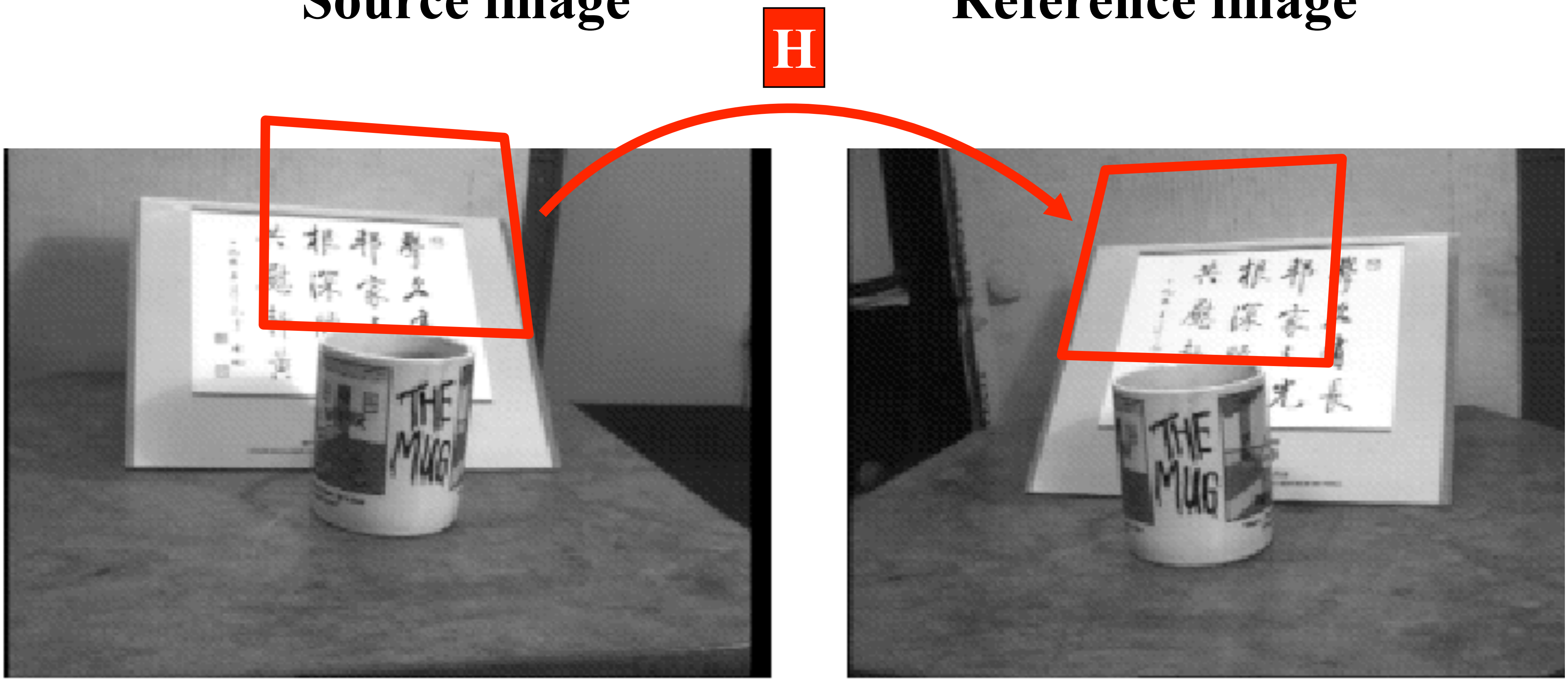
Assumes scene is roughly planar.

What if scene isn't planar?
Alignment will not be good if significant 3D relief

→ "Ghosting"

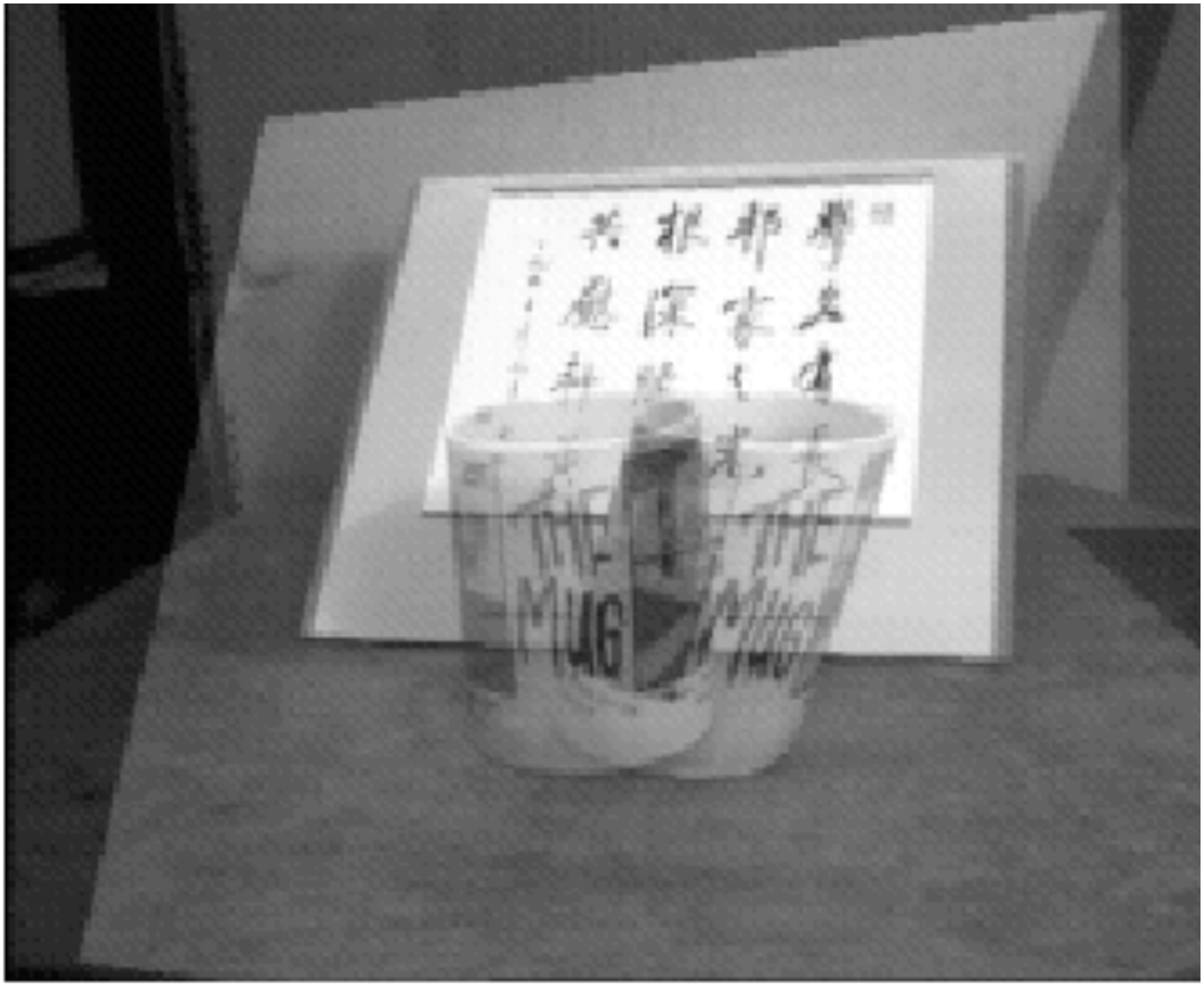
Ghosting Example

Source image **Reference image**



Ghosting Example (cont)

Mosaic



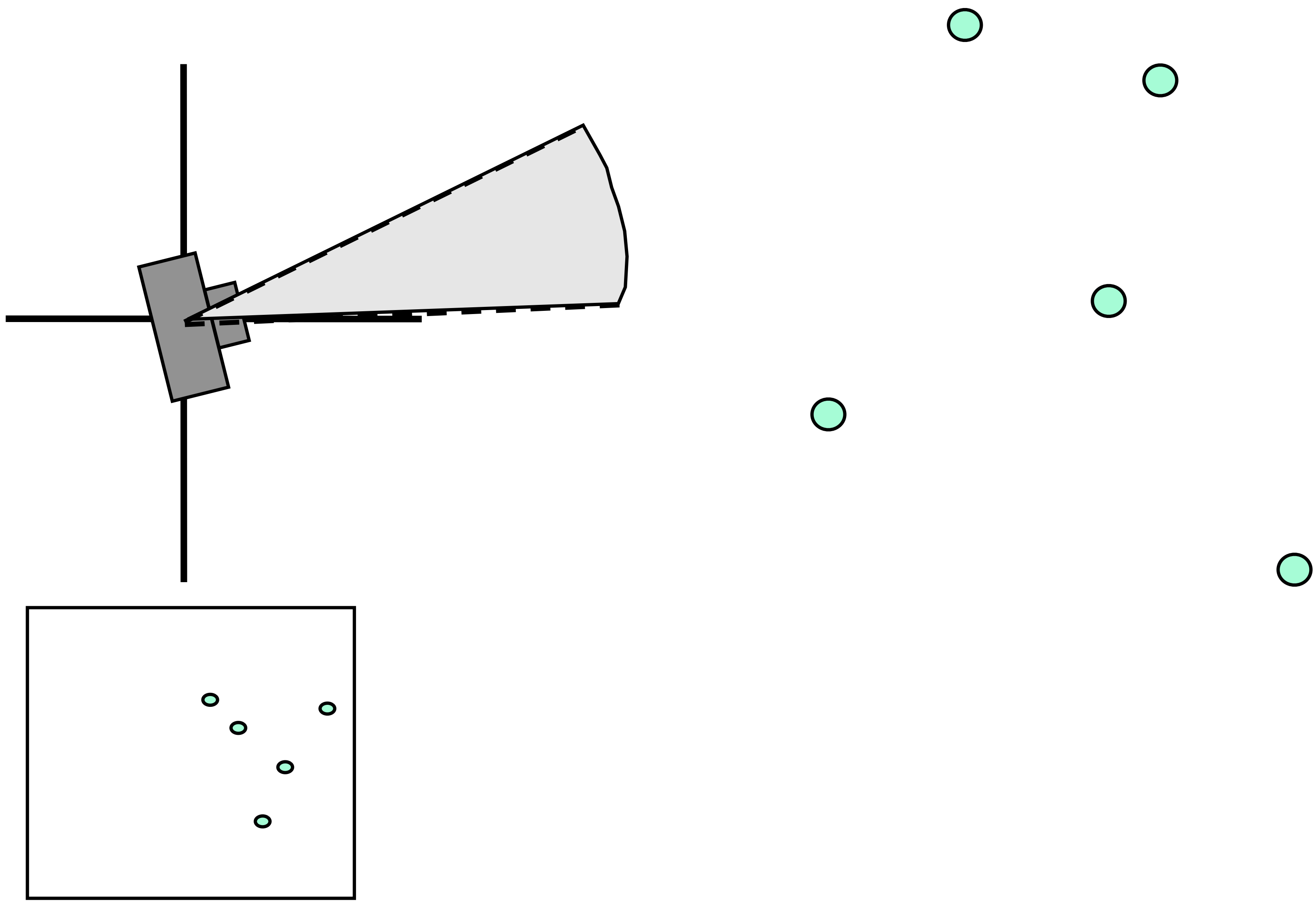
Mosaics from Rotating Cameras

However, there is a mitigating factor in regards to ghosting…

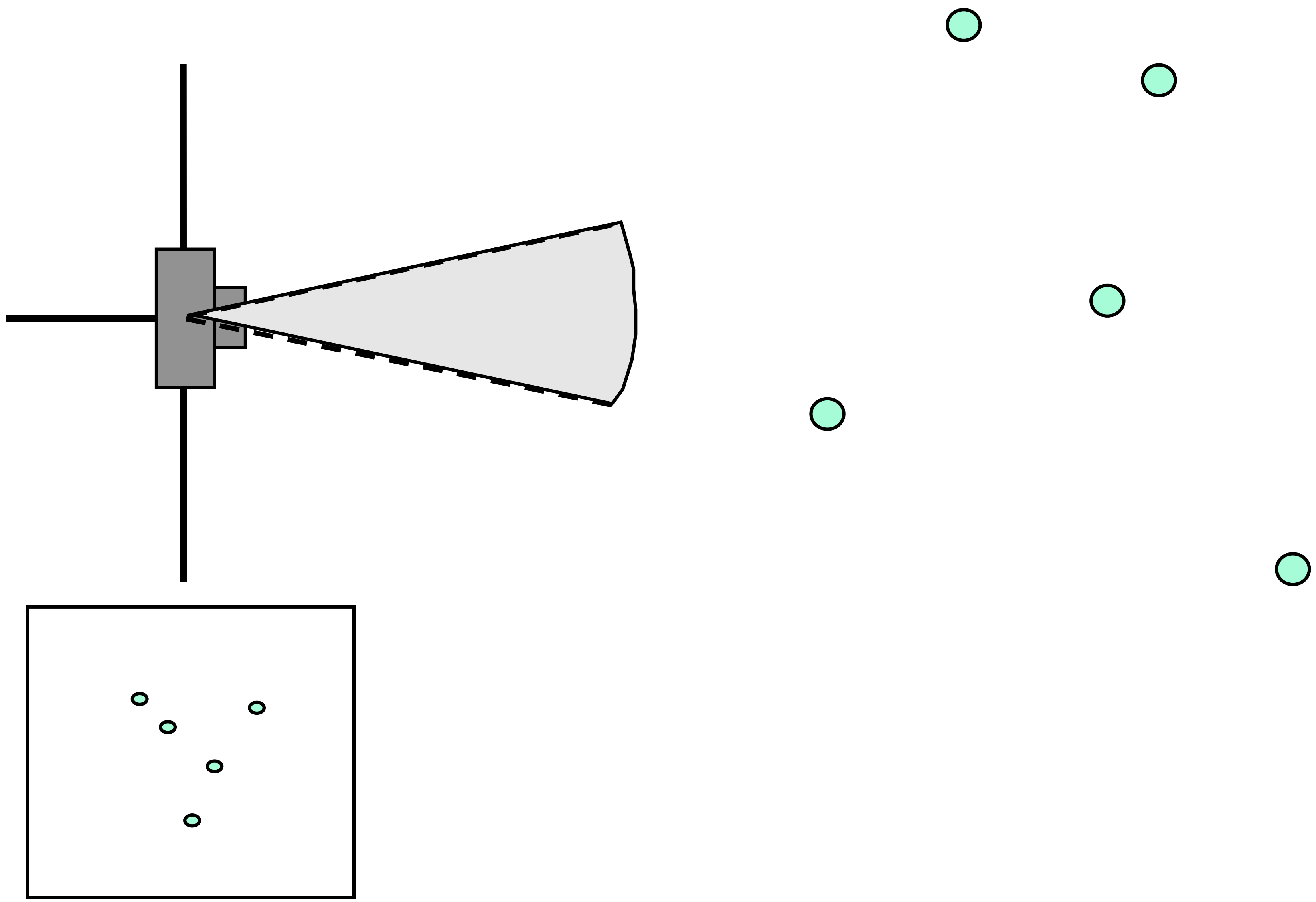
Images taken from a rotating camera are related by a 2D homography…

<u>regardless of scene structure!</u>

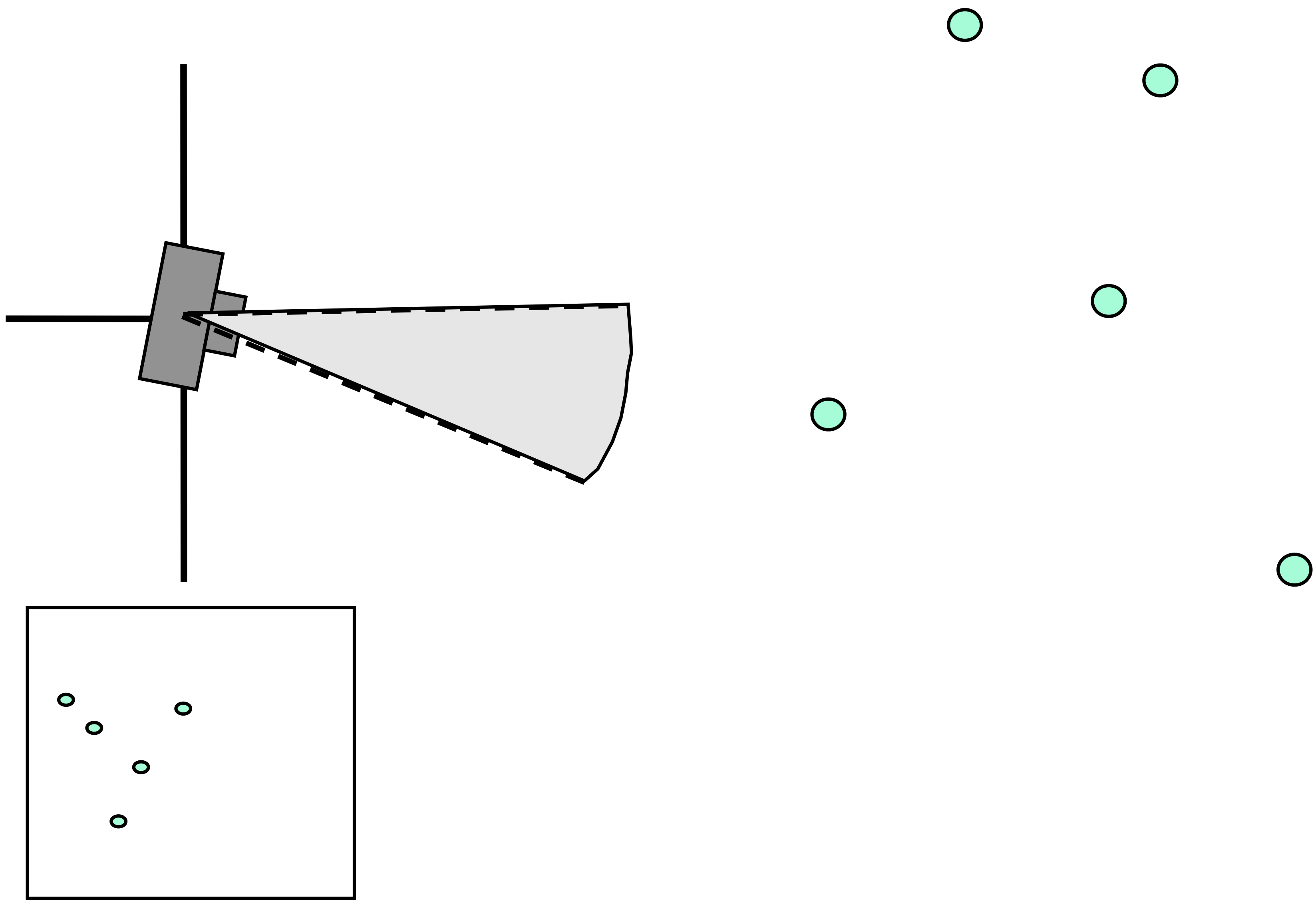
Rotating Camera (top-down view)



Rotating Camera (top-down view)

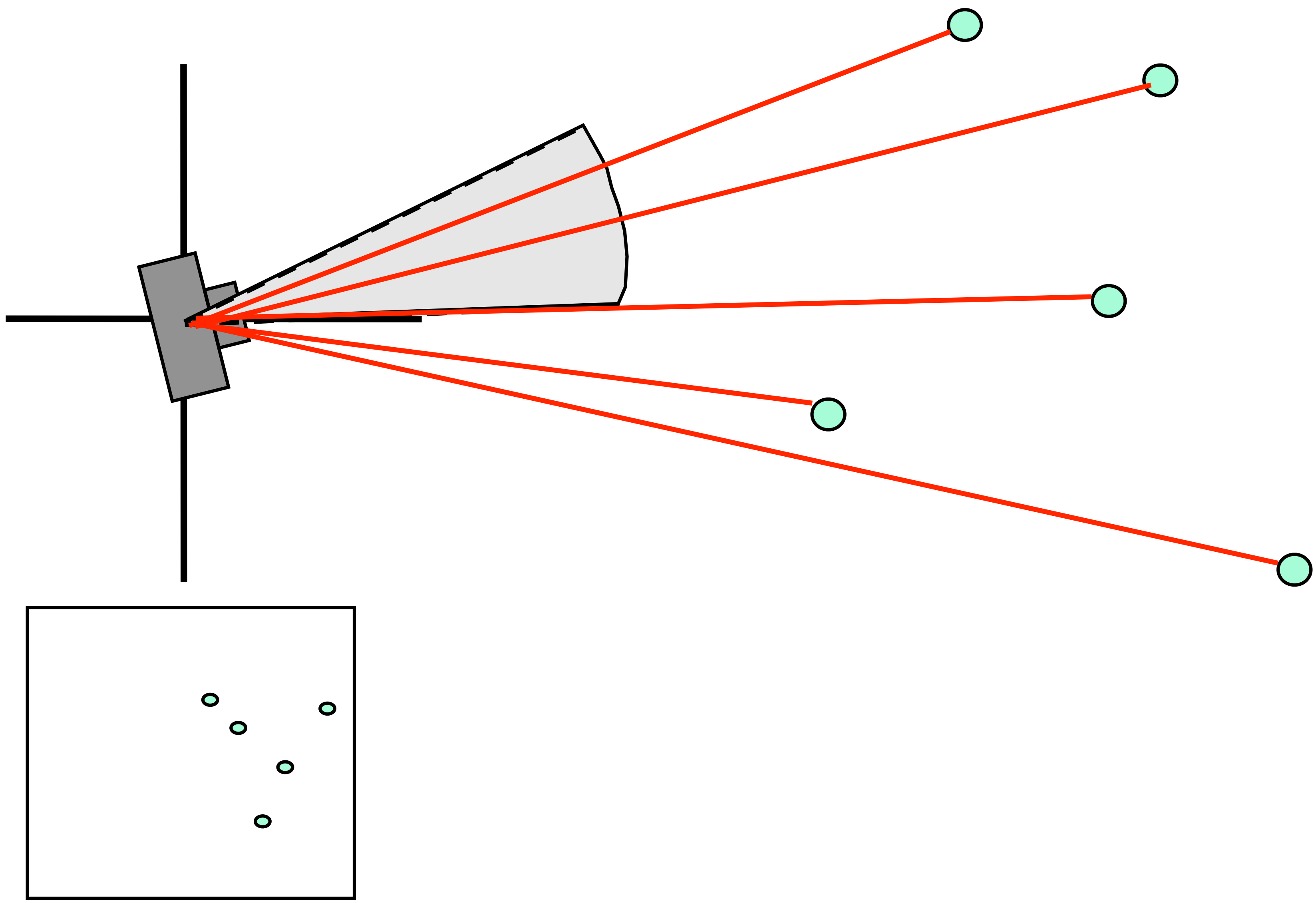


Rotating Camera (top-down view)



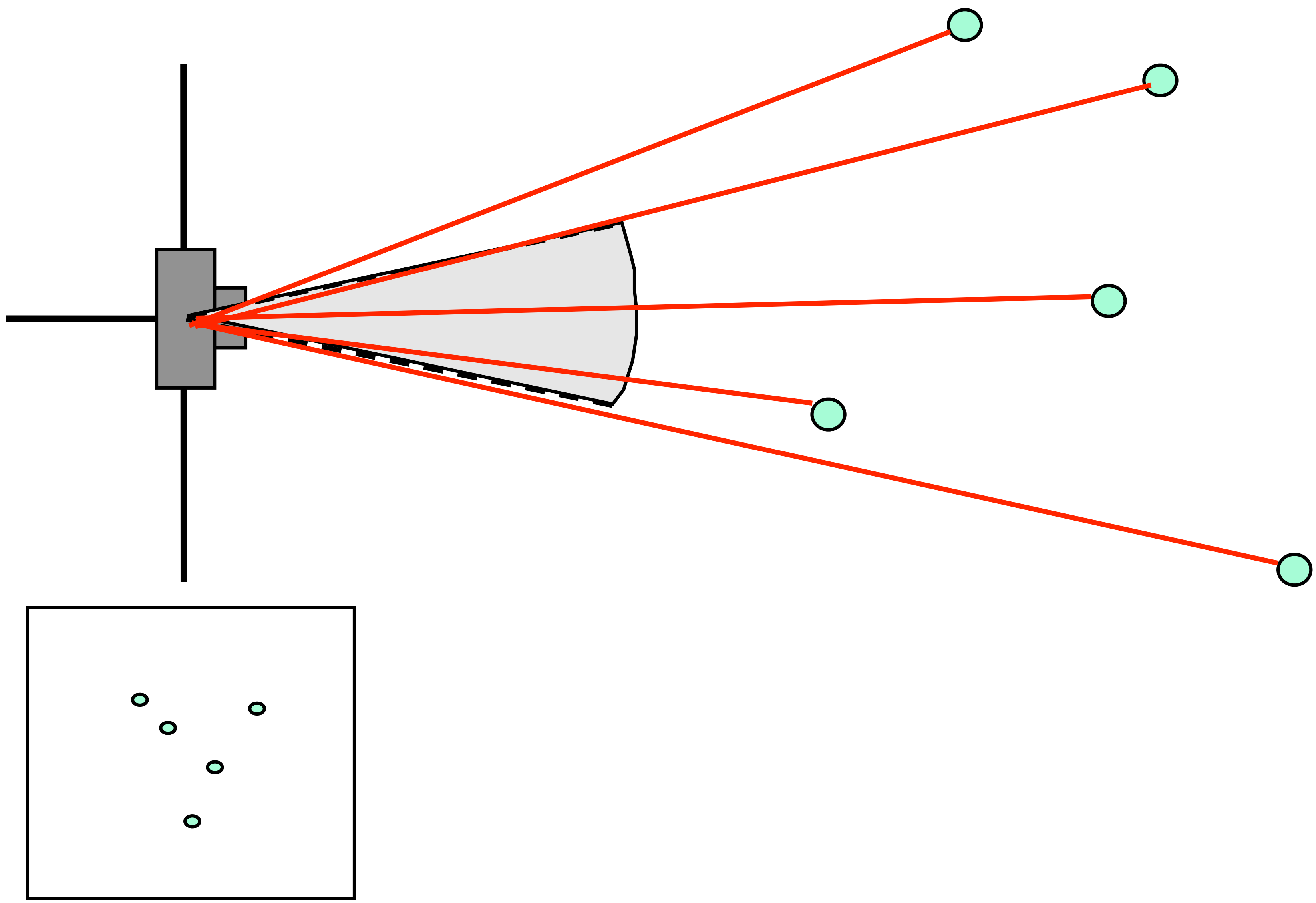
Rotating Camera (top-down view)

Rays in camera coord system are invariant!



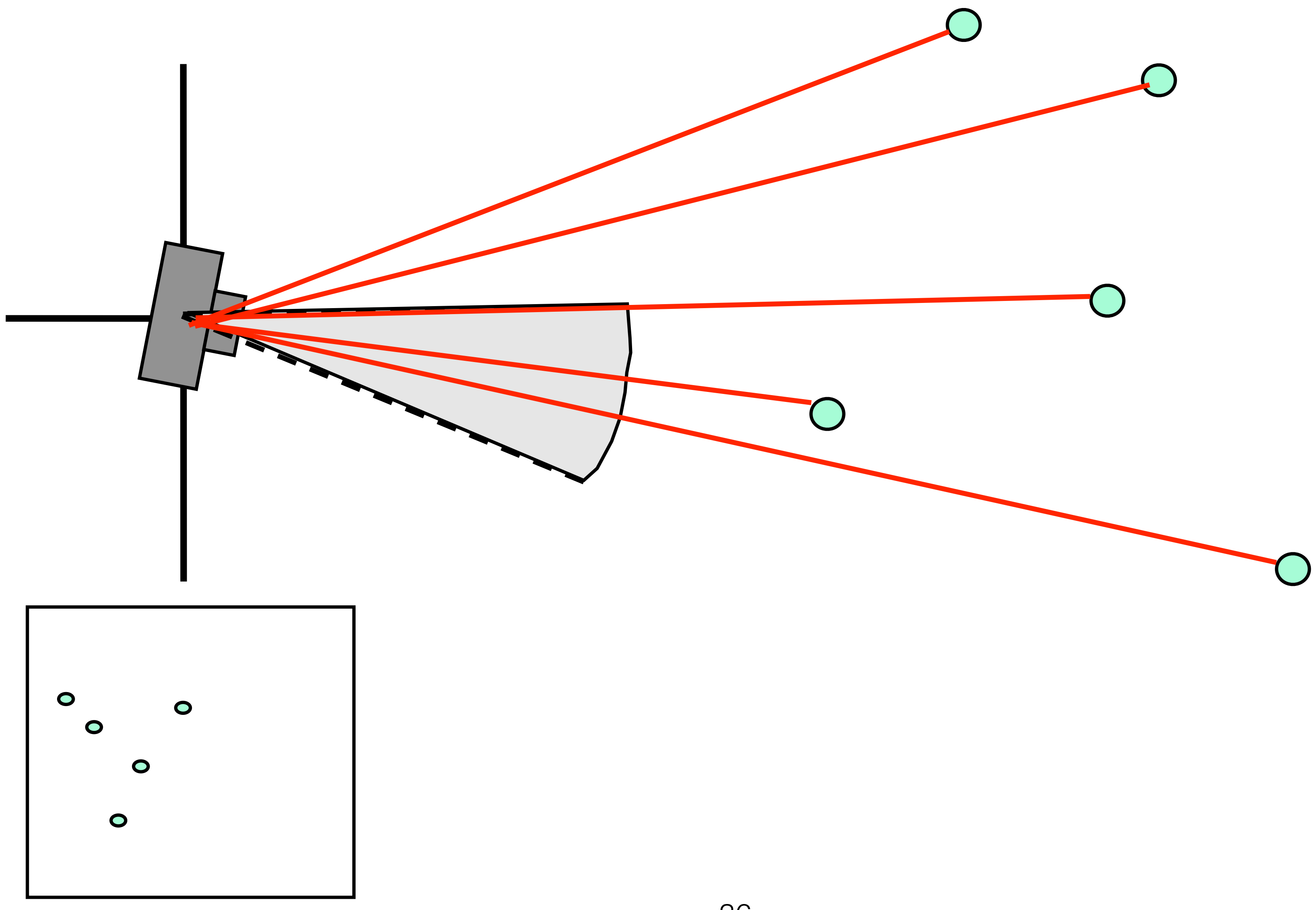
Rotating Camera (top-down view)

Rays in camera coord system are invariant!



Rotating Camera (top-down view)

Rays in camera coord system are invariant!



Special Case : Rotating Camera

Internal params **Projection** **Relative R,T**

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Relative Rotation of camera

Translation is 0
This is important!

Special Case : Rotating Camera

Internal params **Projection** **Relative R,T**

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Internal params}} \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{Projection}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{Relative R,T}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Relations among Images Taken by Rotating Camera

Image 1 $$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Same ray!

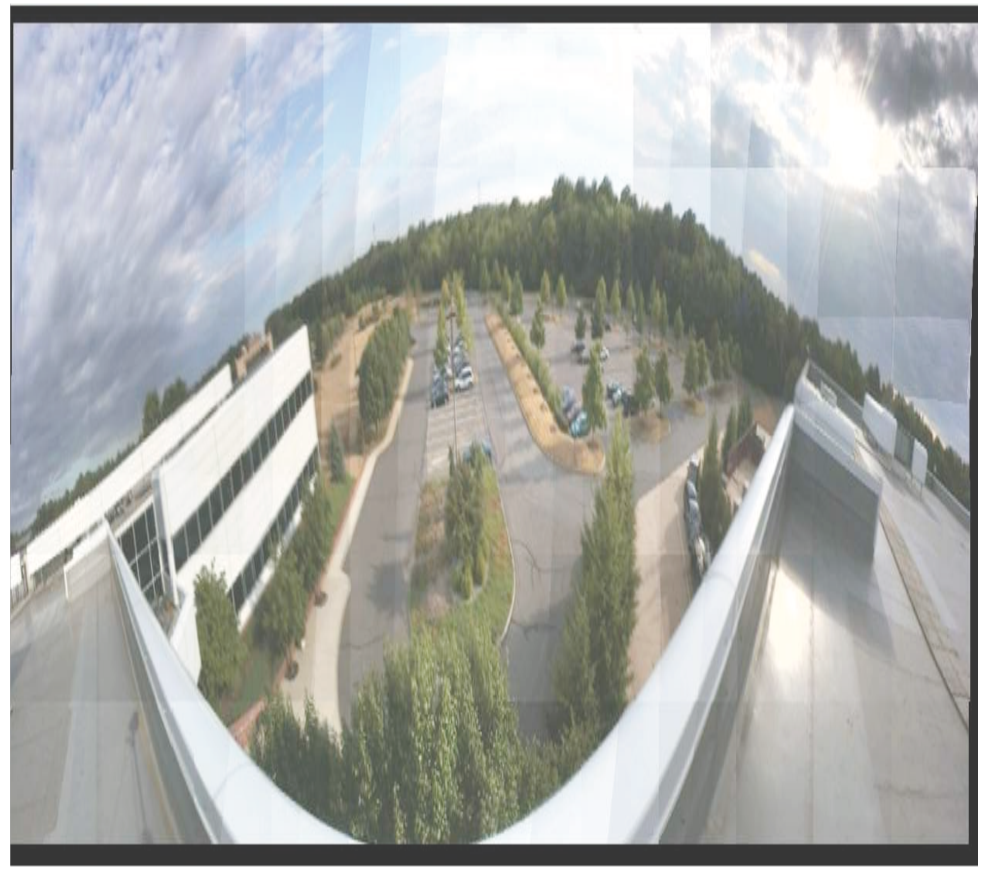
Image 2 $$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & h'_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ h'_{31} & h'_{32} & h'_{33} \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Mosaicing Example

Original Images (from a pan/tilt camera)

One more detail: Blending!



Approaches to Blending

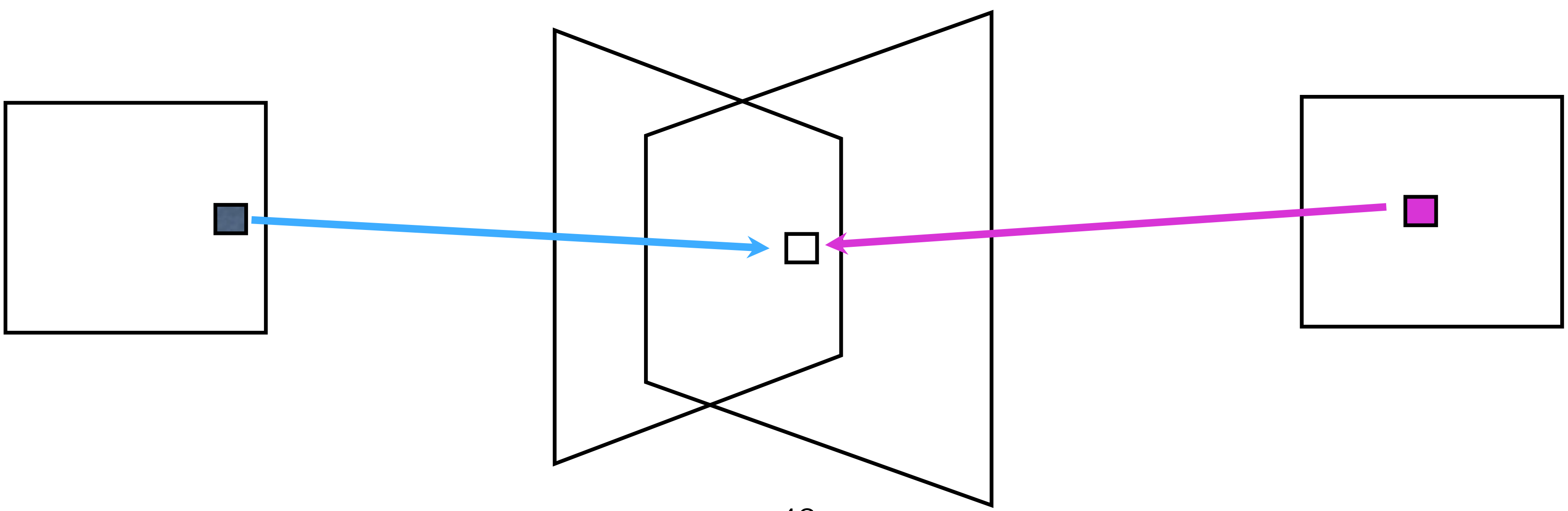
How to combine colors in area of overlap?

1) Straight averaging $\mathbf{P = (P_1 + P_2) / 2}$

2) Feathering $\mathbf{P = (w_1 * P_1 + w_2 * P_2) / (w_1 + w_2)}$

With w_i being distance from image border

3) Equalize intensity statistics (gain, offset)



360 Degree Panoramas?

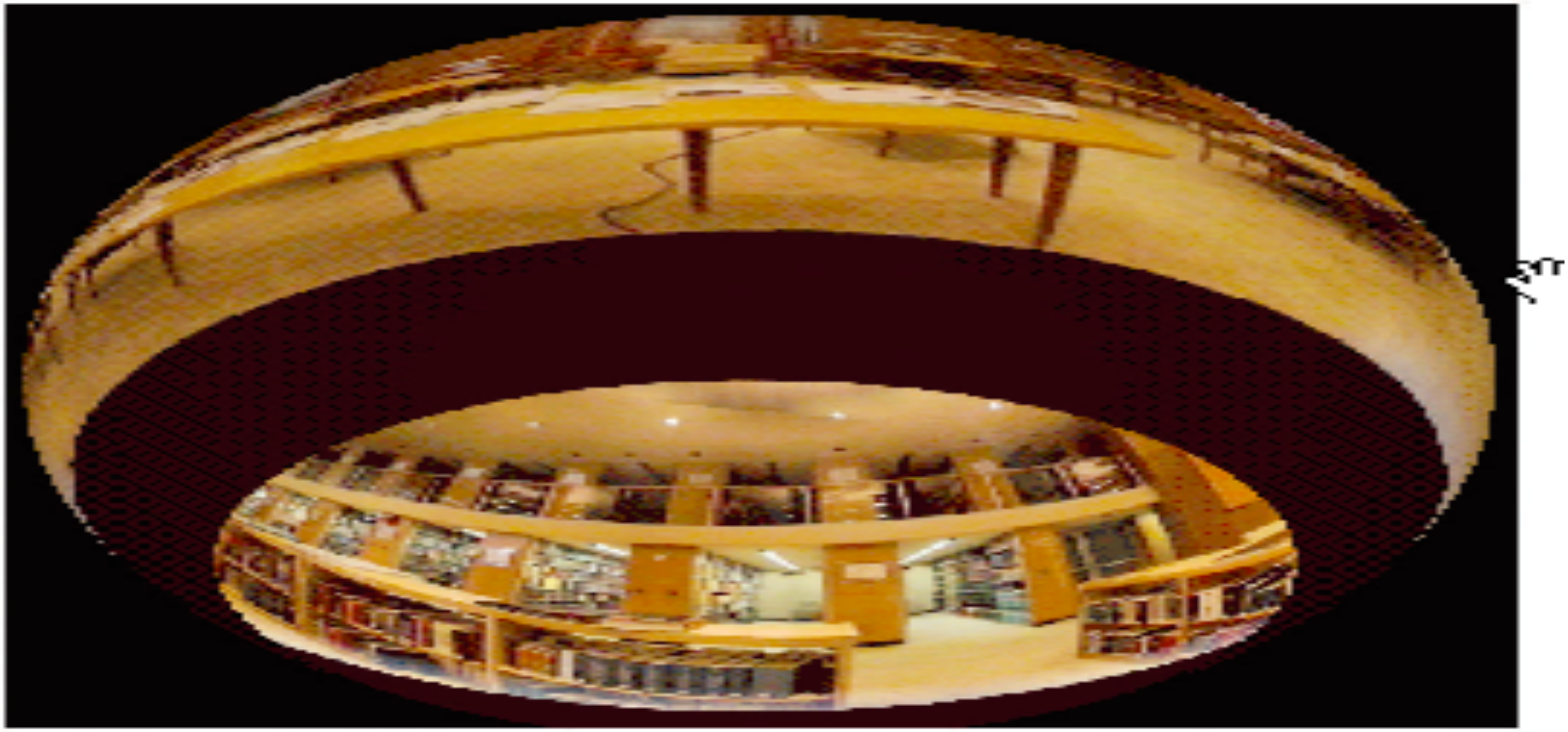
Problem: Can’t just choose a reference image to map all other images to.

Solution: Use cylindrical or spherical mosaic surface rather than a plane.

Panorama Input images

Sarnoff

Spherical Panorama Result



Sarnoff

Panorama Unwarped

Sarnoff

Practical Issues

How to:

- Find four or more point correspondences
- Estimate the homography given the point correspondences.
- (Un)warp image pixel values to produce a new picture.

Feature Matching

Feature Matching

Problem:

Given two images with partial overlap, find interesting points in each image and determine pair-wise point correspondences.

Challenges:

Need to find features that are robust to camera pose changes

Need a way to compare features

Need a method to reject bad correspondences/outliers

Example

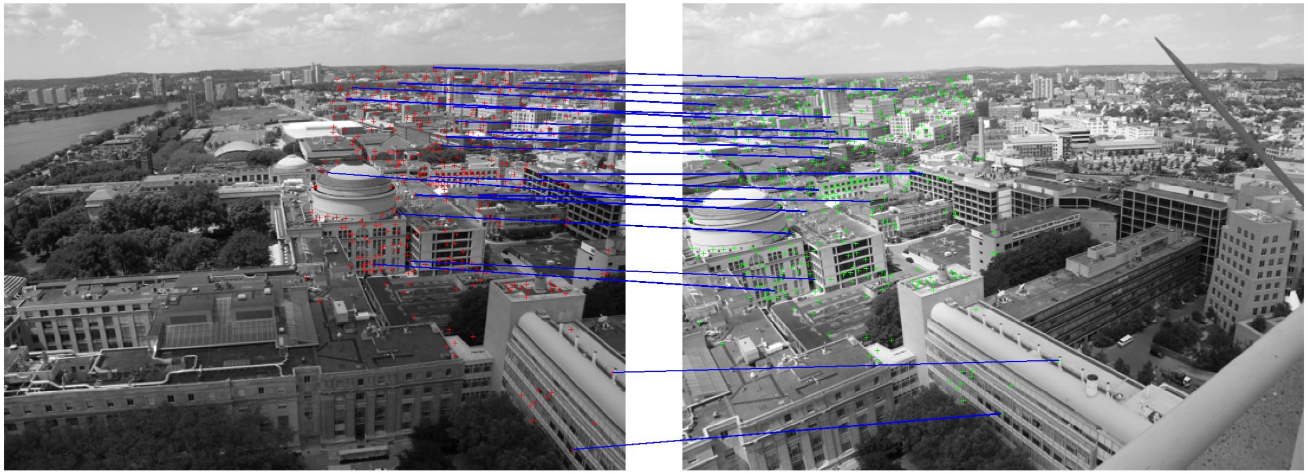
First, find “interesting” features in each image

Example

First, find "interesting" features in each image

Example

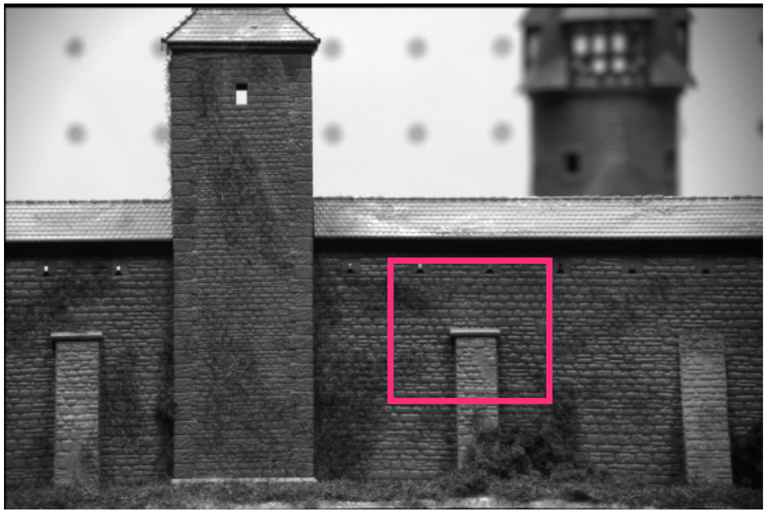
Match features: for each feature in Image 1, find top N best matches in Image 2.

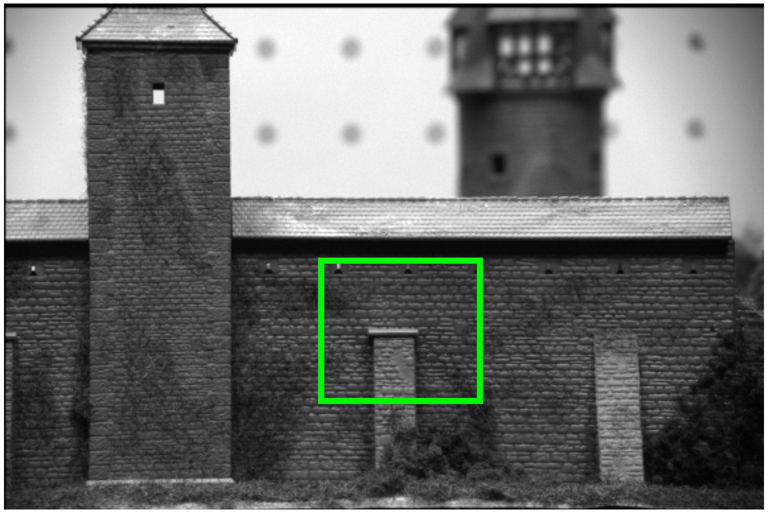


Comparing Features: Correlation-based Algorithms

Elements to be matched:

image WINDOWS of fixed size.





Comparing Windows:

Some possible measures:

 $\stackrel{?}{=}$ 

f g

$$\max_{[i,j]\in R} |f(i,j) - g(i,j)|$$

$$\sum_{[i,j]\in R} |f(i,j) - g(i,j)|$$

$$SSD = \sum_{[i,j]\in R} (f(i,j) - g(i,j))^2$$

$$C_{fg} = \sum_{[i,j]\in R} f(i,j)g(i,j)$$

Most popular

SSD and Cross-Correlation C_{fg}

Cross correlation is closely related to the SSD:

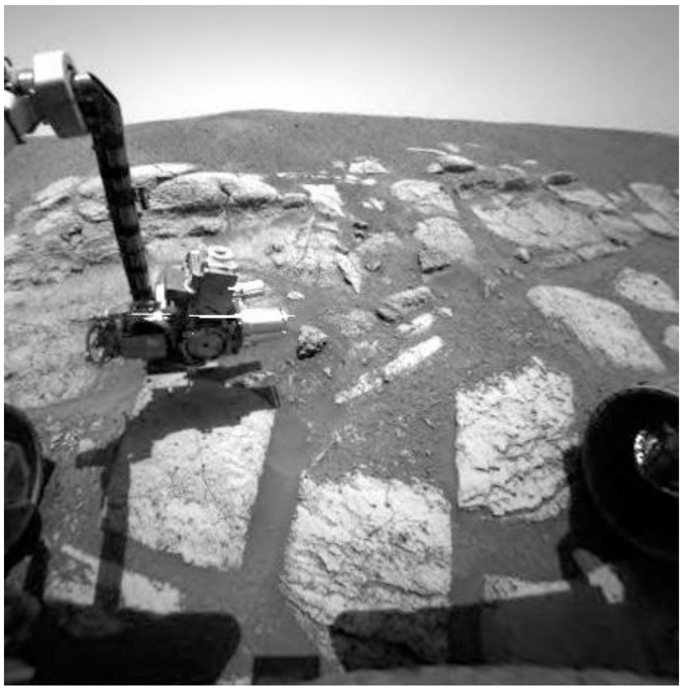
$$C_{fg} = \sum_{[i,j]\in R} f(i,j)g(i,j)$$

$$SSD = \sum_{[i,j]\in R} (f-g)^2 = $$

$$= \sum_{[i,j]\in R} f^2 + \sum_{[i,j]\in R} g^2 - 2\sum_{[i,j]\in R} fg$$

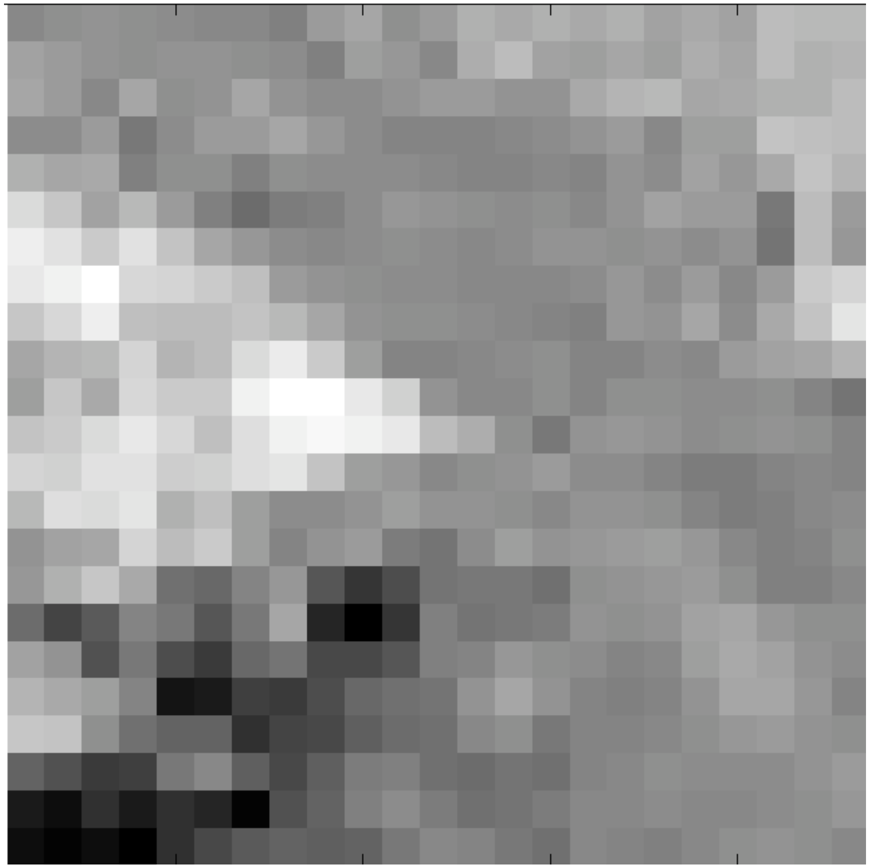
Large cross correlation ~ Small SSD

Example



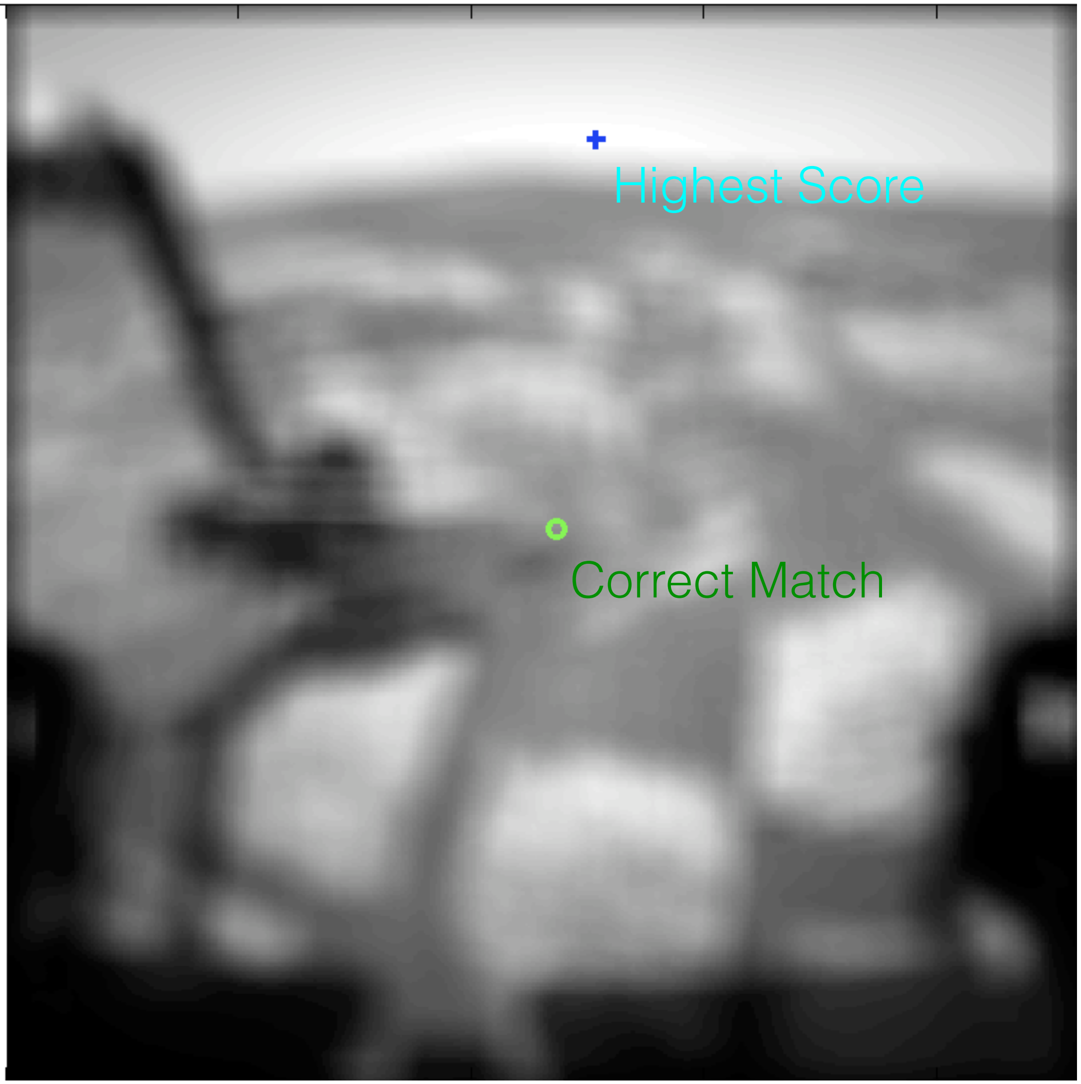
Stereo pair "El Capitan" from NASA Mars Rover mission

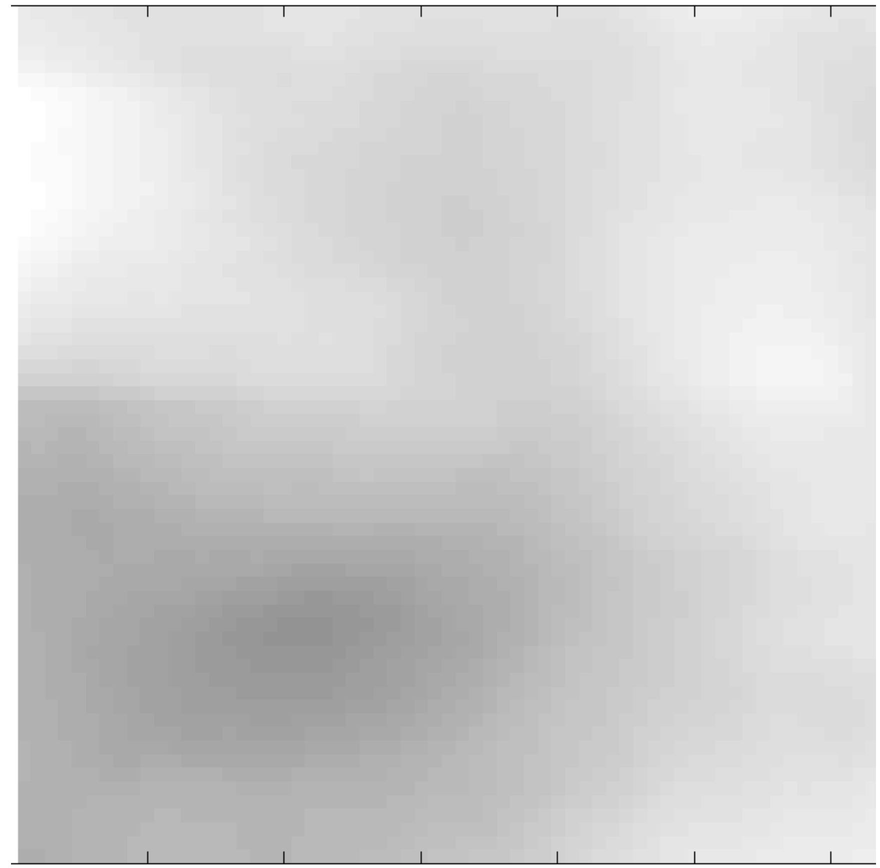
Example



template

Example: cross-correlation score = imfilter(image2,template)





Score Around Correct Match

Problems with Pure Cross-Correlation

Consider the following two patches:

a	b	c
d	e	f
g	h	i

v	v	v
v	v	v
v	v	v

Cross-correlation = (a + b + c + d + e + … + i) . v

Problems with Pure Cross-Correlation

Now consider a brighter patch:

a	b	c
d	e	f
g	h	i

2v	2v	2v
2v	2v	2v
2v	2v	2v

Cross-correlation = (a + b + c + d + e + … + i) . 2v

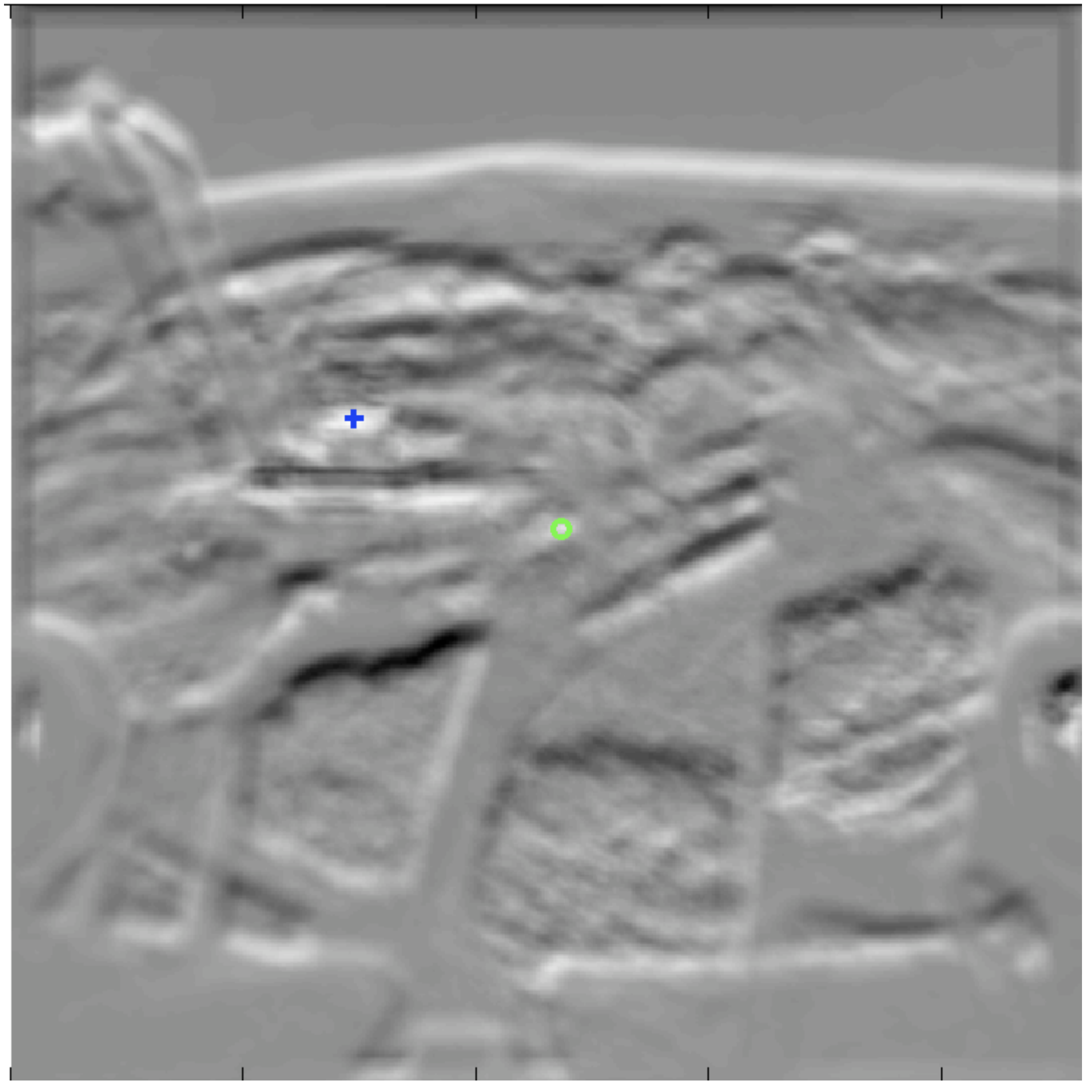
Larger score, regardless of what the template is!

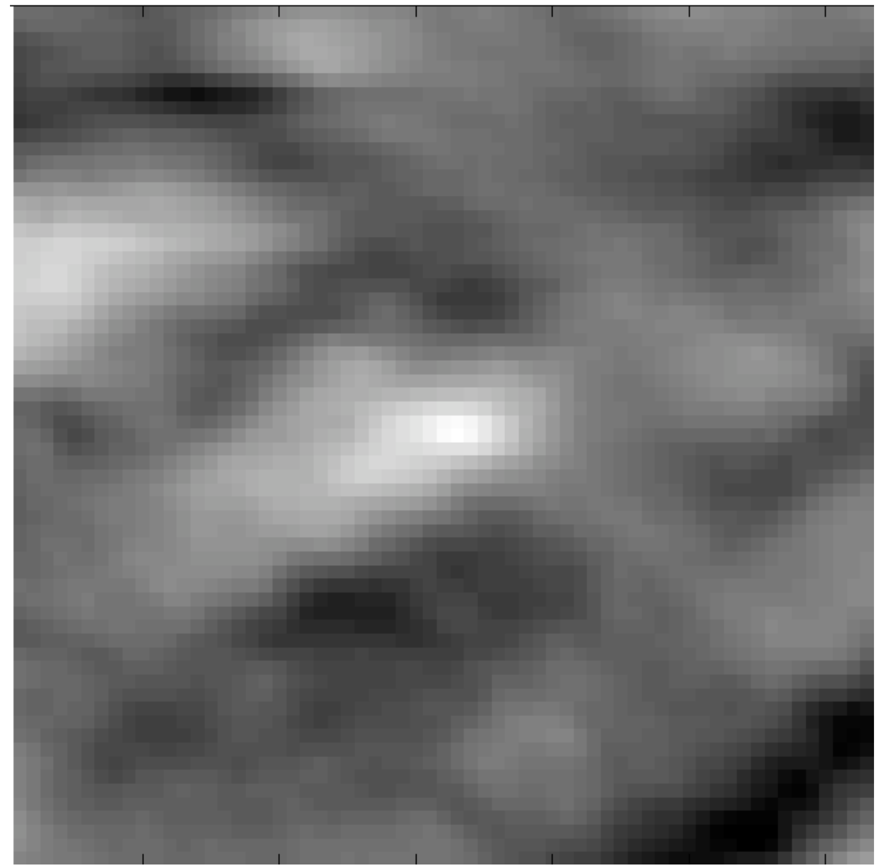
Solution

Subtract off the mean value of the template

In this way, the correlation score is higher ONLY when darker parts of the template overlap with darker parts of the image, and brighter parts of the template overlap brighter parts of the image.

Correlation, zero-mean template





Better … but still the highest score is not the correct match.

Handling Intensity Changes

Intensity changes:

The camera taking the second image might have different intensity response characteristics than the camera taking the first image

Illumination in the scene might change

The camera might have auto-gain control set, so its response changes as the camera moves through the scene.

Handling Intensity Changes

One approach could be to estimate the change in intensity and compensate for it.

A second approach is to use **normalized** matching functions that are invariant to intensity changes.

Normalization

•When a scene is imaged by different sensors, or under different illumination intensities, both the SSD and the C_{fg} can be large for windows representing the same area in the scene!

•A solution is to **NORMALIZE** the pixels in the windows before comparing them (take away the mean and scale with the std. dev):

$$\hat{f} = \frac{f}{||f||} = \frac{f}{\sqrt{\sum_{[i,j]\in R} f^2(i,j)}}$$

$$\hat{g} = \frac{g}{||g||} = \frac{g}{\sqrt{\sum_{[i,j]\in R} g^2(i,j)}}$$

Normalized Cross-Correlation N_{fg}

$$N_{fg} = C_{\hat{f}\hat{g}} = \sum_{[i,j]\in R} \hat{f}(i,j)\hat{g}(i,j)$$

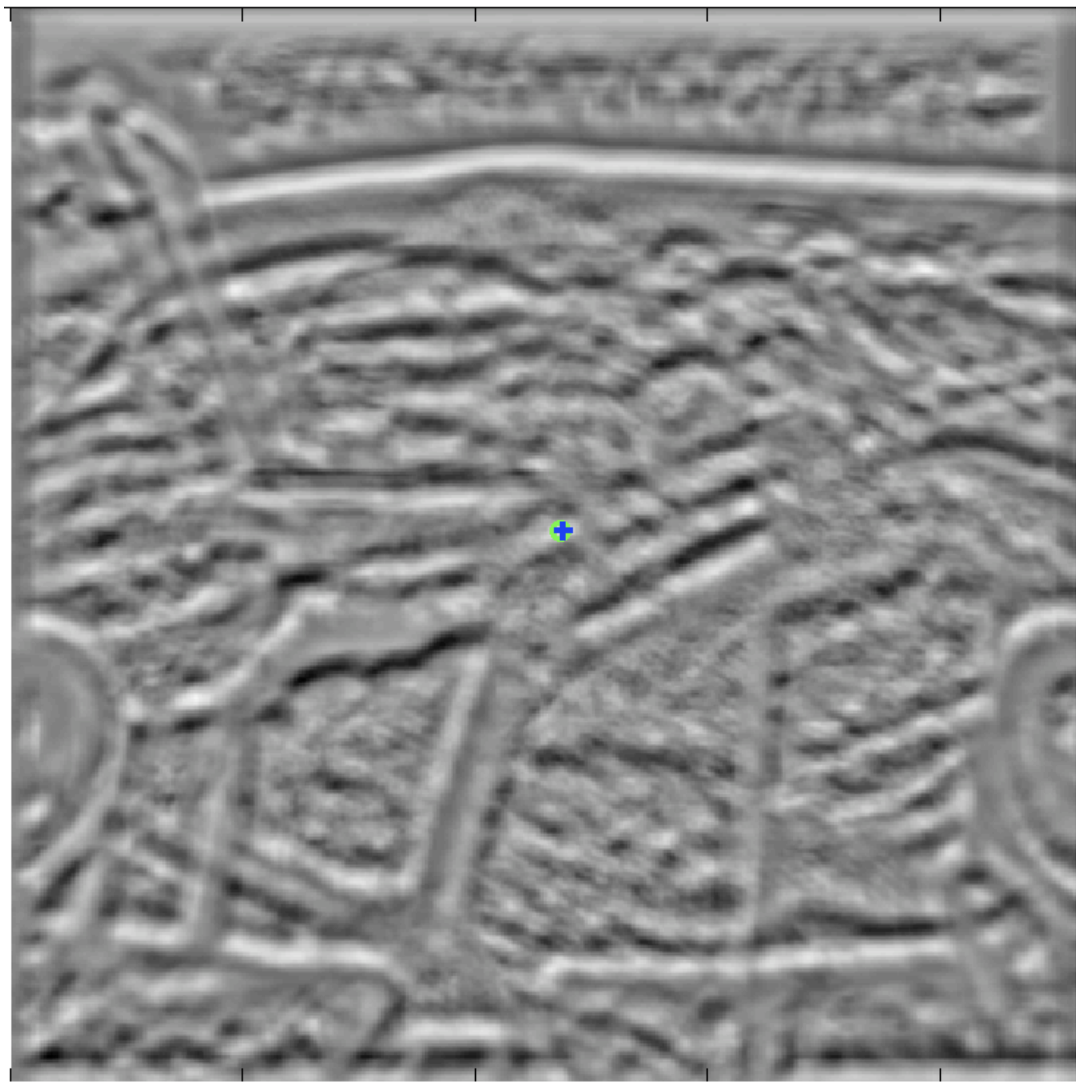
$$\hat{f} = \frac{f}{||f||} \quad \hat{g} = \frac{g}{||g||}$$

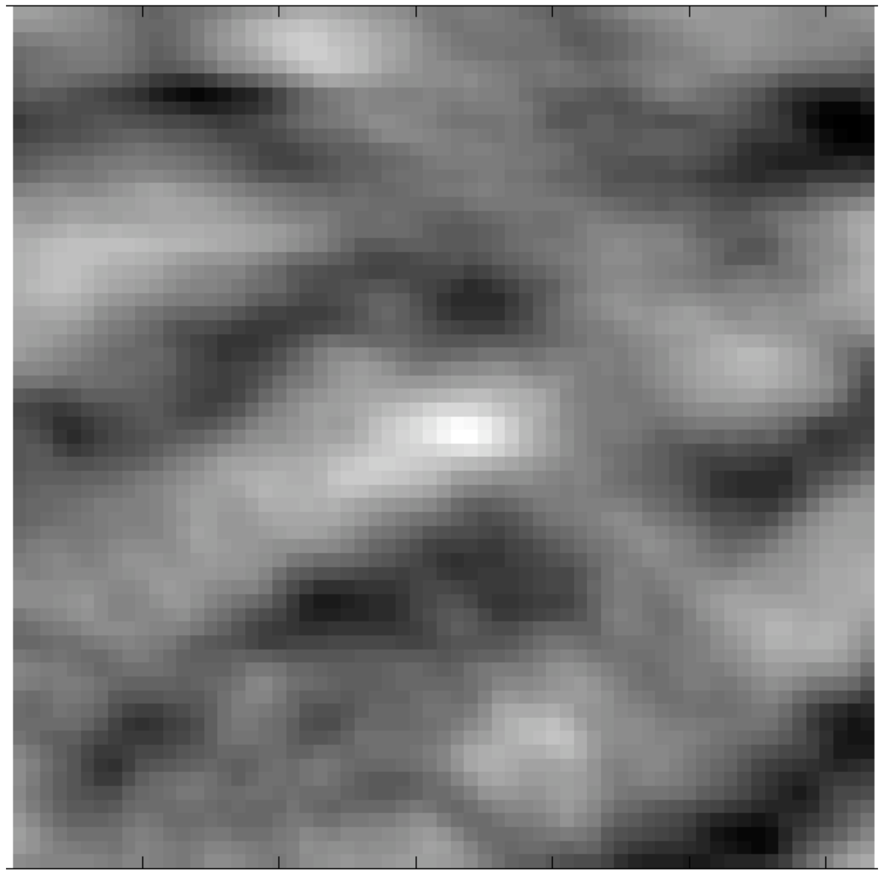
It is EQUIVALENT to the (normalized) SSD:

$$NSSD = \sum_{[i,j]\in R} (\hat{f} - \hat{g})^2 = \sum_{[i,j]\in R} (\hat{f}^2 + \hat{g}^2 - 2\hat{f}\hat{g})$$

$$= 1 + 1 - 2 \sum_{[i,j]\in R} \hat{f}\hat{g} = 2 - 2N_{fg}$$

Normalized Cross Correlation





Highest score coincides with best match. Also, it looks less likely that we could get the wrong match.

Normalized Cross Correlation

Note that the NCC score ranges between:

-1(completely uncorrelated) and 1(perfect match).

Intuitively:
Each patch is a unit norm vector, and the NCC is their dot-product (cosine of their angle).

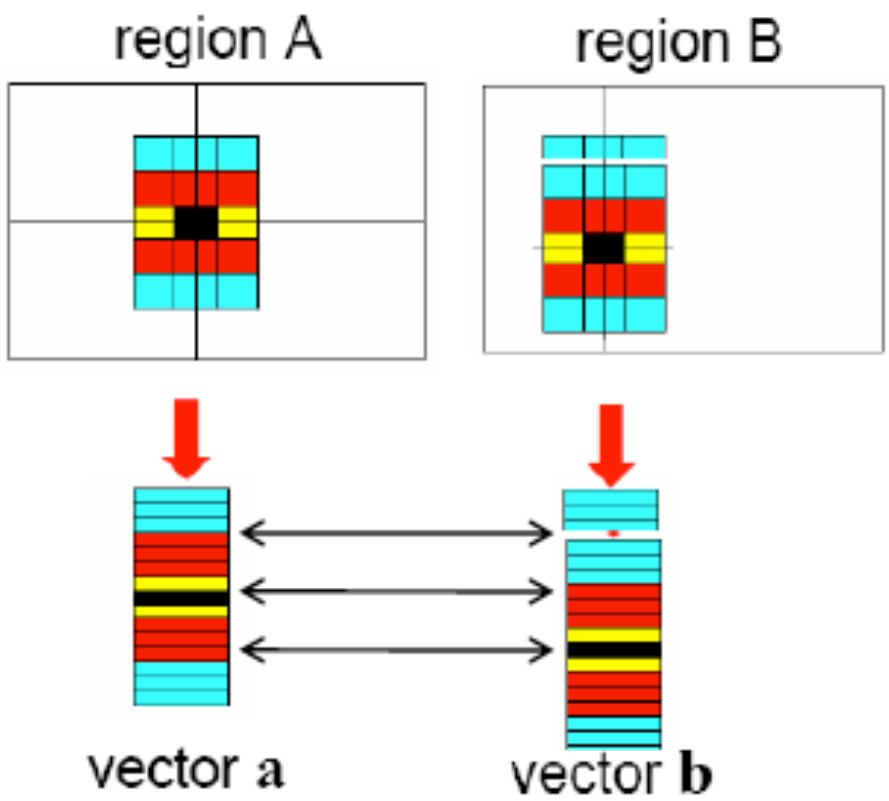
Descriptors: Geometric Transformations

e.g. scale, translation, rotation

Descriptors: Photometric Transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Raw Patches as Local Descriptors

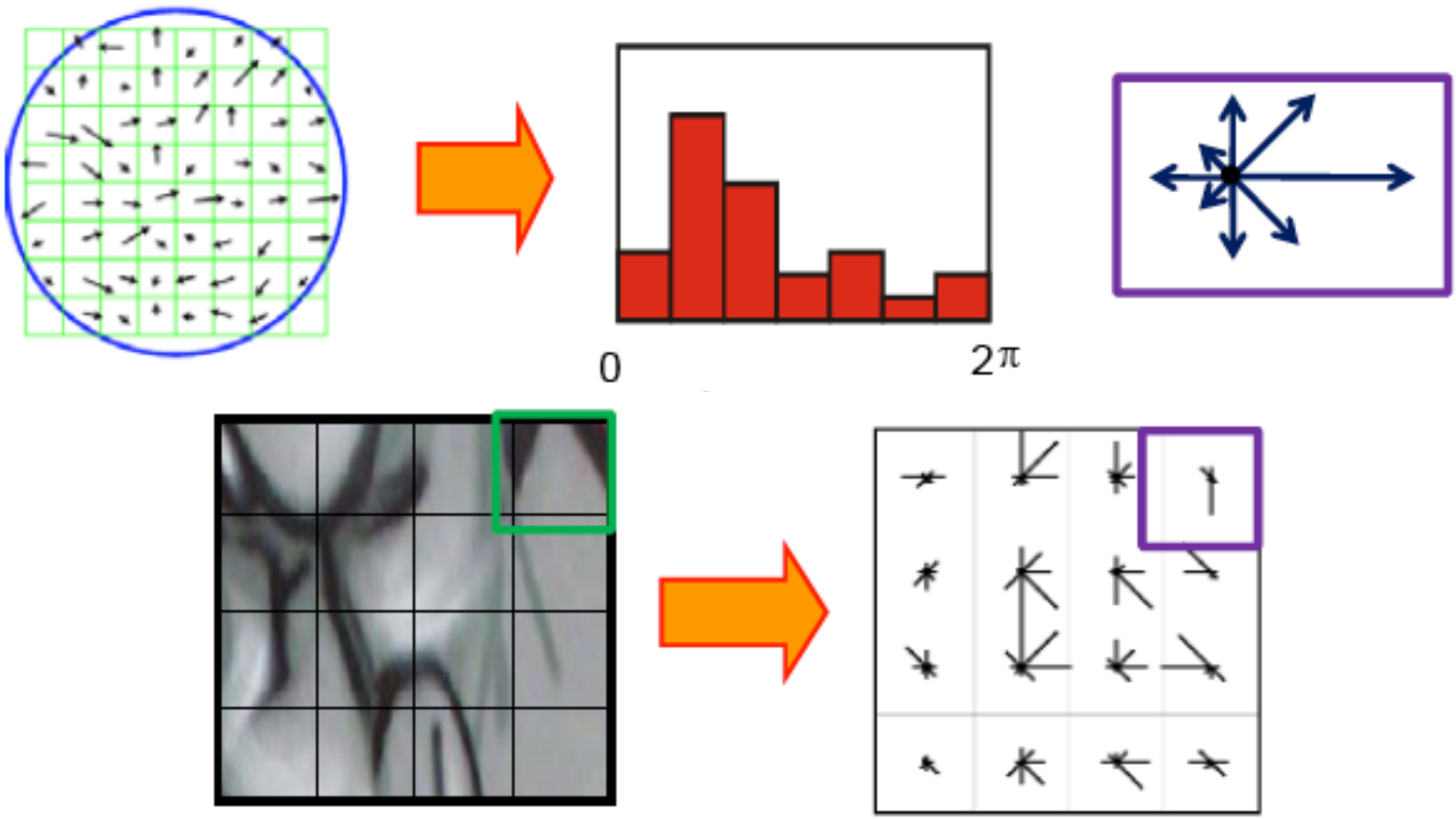


The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

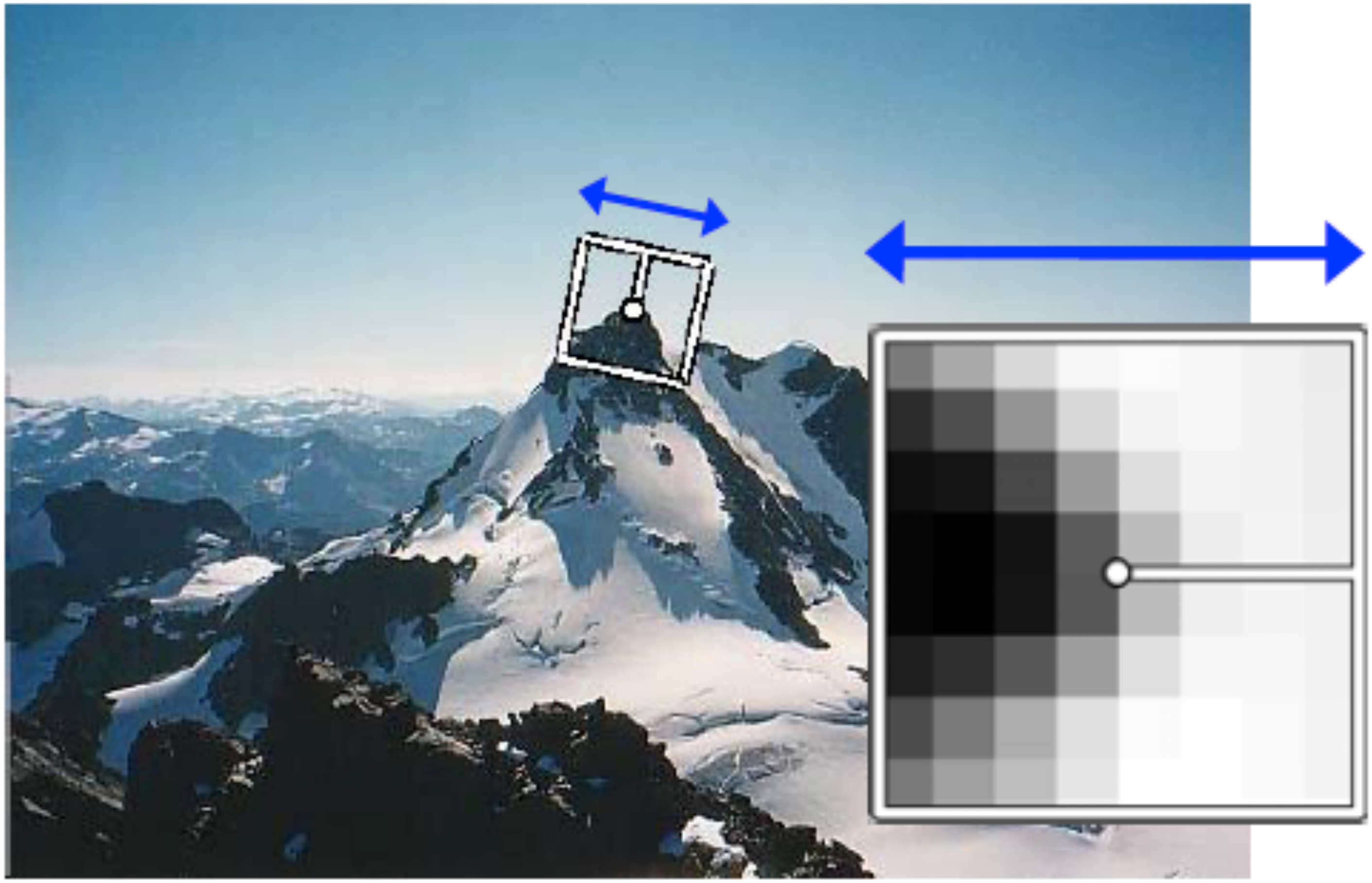
But this is very sensitive to even small shifts an rotations

SIFT DESCRIPTOR (Lowe 2004)

Use histograms to bin pixels within sub-patches according to their orientation.



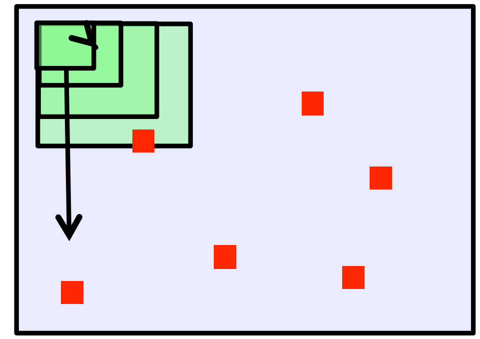
Rotation Invariance

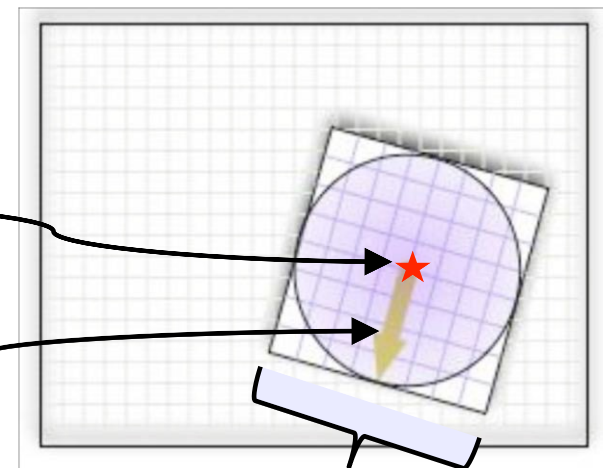


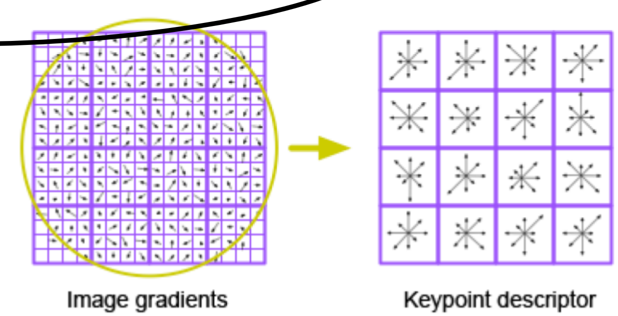
Rotate patch according to its dominant gradient orientation: “canonical” representation.

SIFT: Scale Invariant Feature Transform

- **Detection stage for SIFT features:**
  - **Scale-space extrema detection.**
    - ➢ **Convolve image with template at 4 different scales.**
    - ➢ **Difference of Gaussians template picks out location with lots of gradient changes.**
    - ➢ **Get both scale invariance and u-v locations of points.**
  - **Keypoint localization.**
    - ➢ **Do sub-pixel u-v localization of points.**
  - **Orientation Assignment.**
    - ➢ **Find major gradient direction(s).**
    - ➢ **Get rotation invariance.**
  - **Generation of keypoint descriptors.**
    - ➢ **Based on scale, orientation assignment, pick out size and rotation of local support region.**
    - ➢ **Divide into 4x4 sub-regions and compute 8-bin histogram per region.**
    - ➢ **Concatenate histograms to form 128 element keypoint descriptor.**







SIFT Matching Method

For each SIFT feature found in image 1, match against all features in image 2.

- Compute dot-products of 128 element descriptor.
- Higher dot-product indicates better match.
- Decide to keep a match if it is unique.
  - Use ratio of second best dot product to best.
  - Keep if ratio below 0.6 (heuristic value, can be changed).

Advantages of SIFT descriptor

- Selects interesting points that are highly distinctive.
- Easy to extract and match against a large database of features.
- Invariant/tolerant to noise, scale, rotation, illumination.

Practical limitations.

- Can handle about 30 degrees out-of-image plane pose change before breaking down.

Have some matches: now what?

Might still have bad correspondences, outliers.

Need method to score quality of match.

- Can score match quality individually or by its consensus with other matches.
- Consensus typically works better, more statistics used.

How to measure consensus between matches?

- Can fit a model to the data points.

Two widely used techniques:

- RANdom SAmple Consensus (RANSAC).
- Hough Transform.

Geometric Fitting Models

Use RANSAC with an geometric fitting model.

Homography Model

- For general 3D scene, holds for only small pose changes.
- In case of dominant planar region in image, holds for most pose changes.
- In case of pure rotation, holds for all 3D scenes.

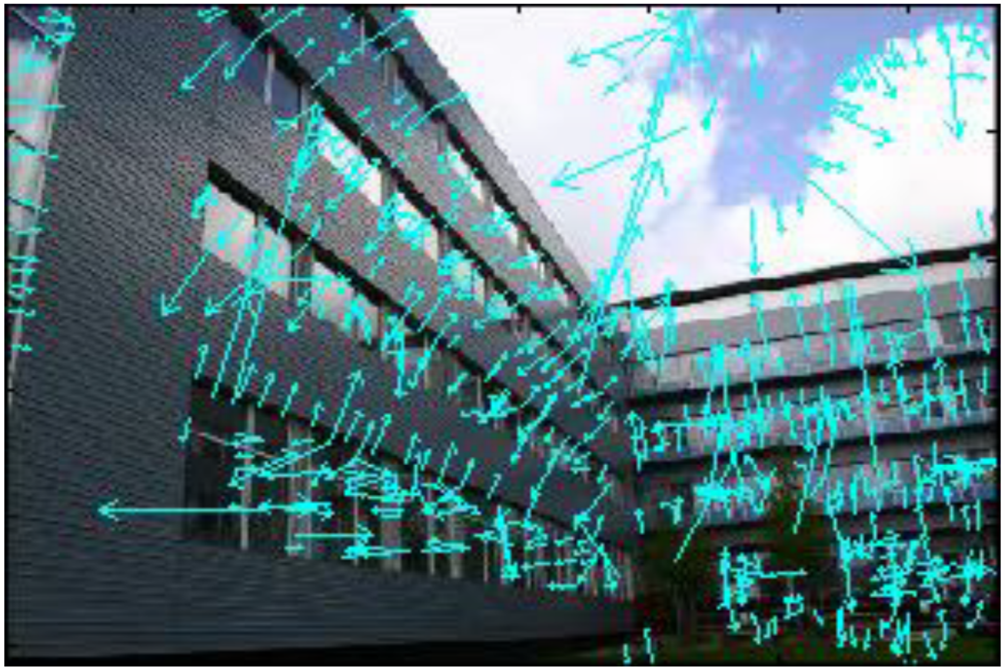
SIFT + RANSAC for Homography with Pure Camera Rotation

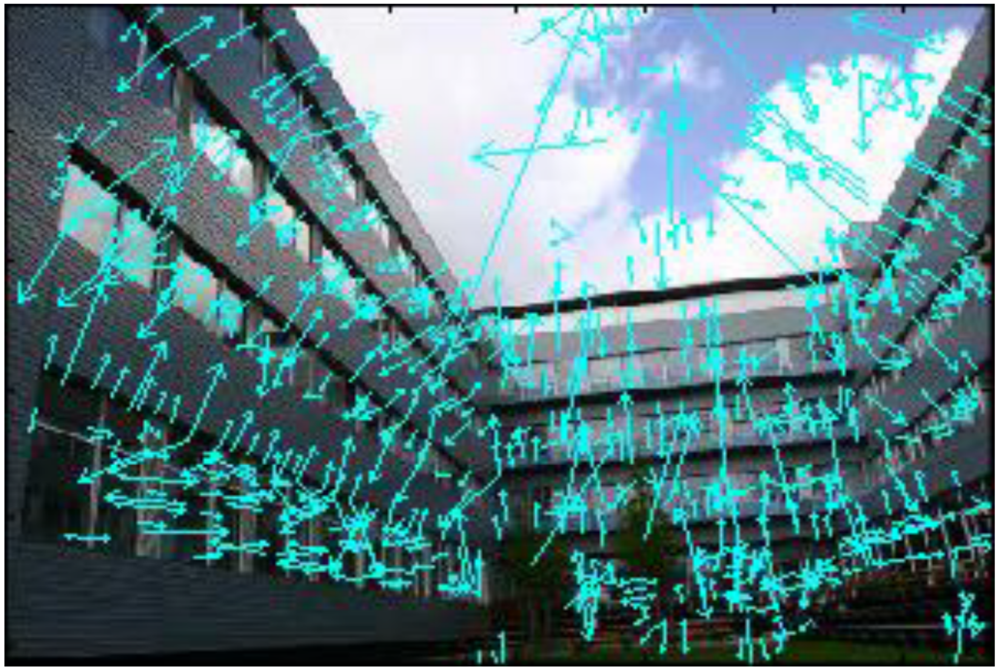
Choose 2 overlapping images.

SIFT + RANSAC for Homography with Pure Camera Rotation

Choose 2 overlapping images.

Find SIFT features for each image.



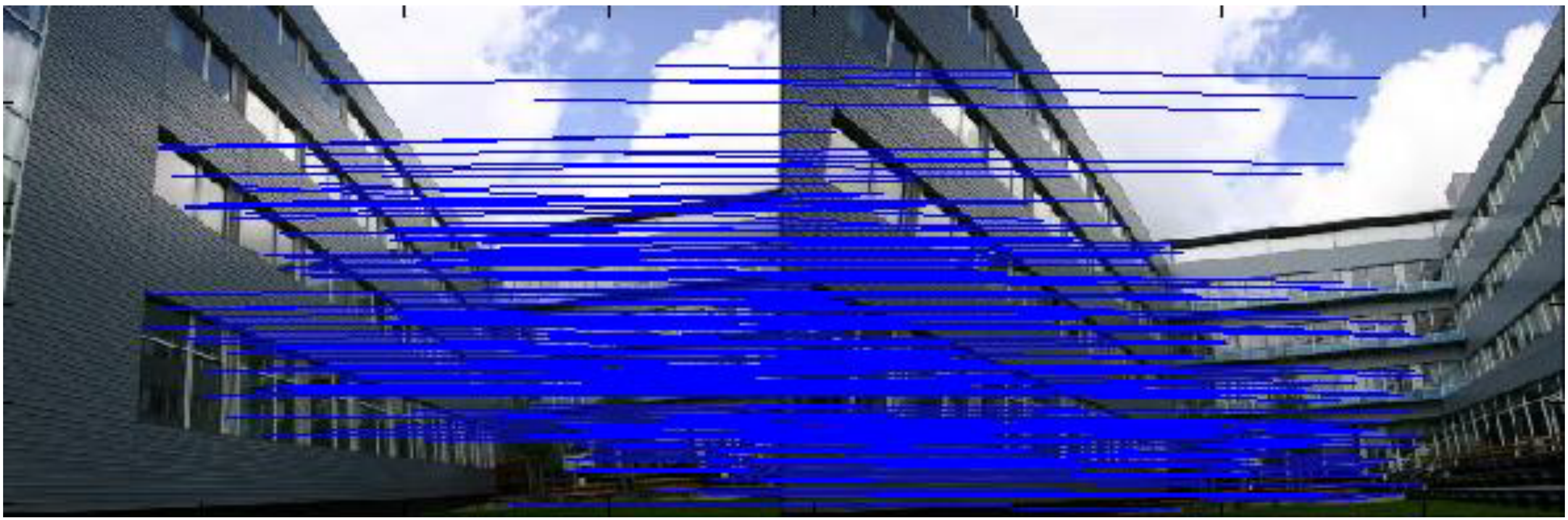


SIFT + RANSAC for Homography with Pure Camera Rotation

Choose 2 overlapping images.

Find SIFT features for each image.

Match SIFT features to get initial point correspondences.



SIFT + RANSAC for Homography with Pure Camera Rotation

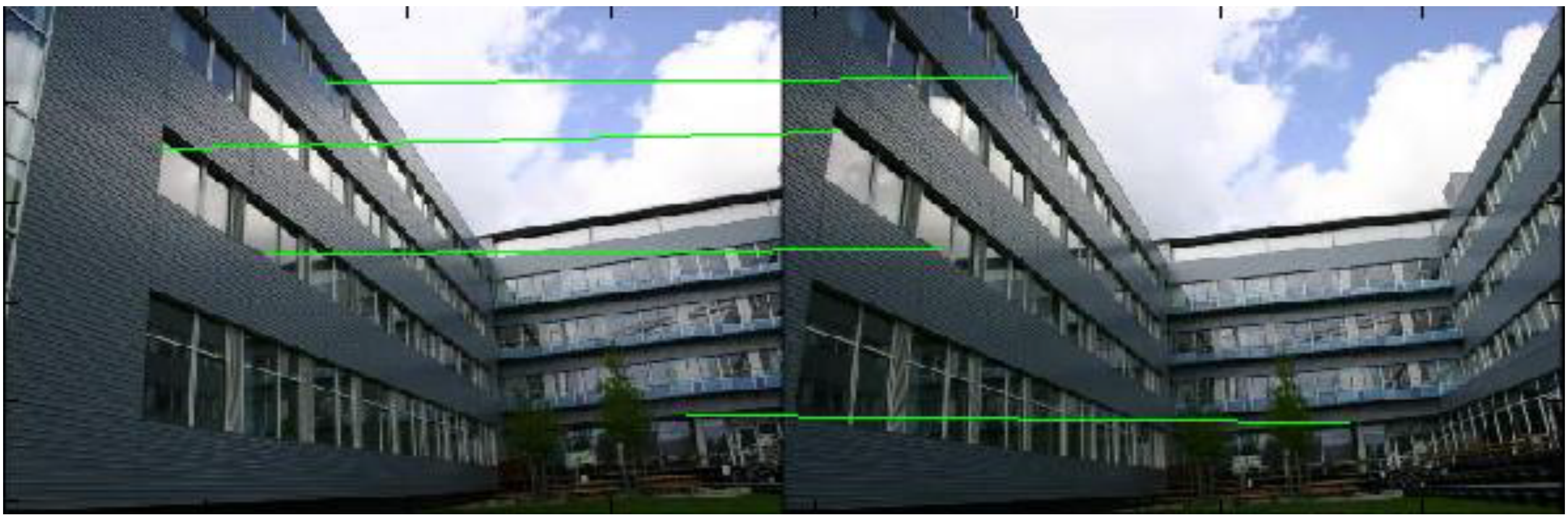
Choose 2 overlapping images.

Find SIFT features for each image.

Match SIFT features to get initial point correspondences.

Run RANSAC:

1. Select minimal number of points (4), find homography.



SIFT + RANSAC for Homography with Pure Camera Rotation

Choose 2 overlapping images.

Find SIFT features for each image.

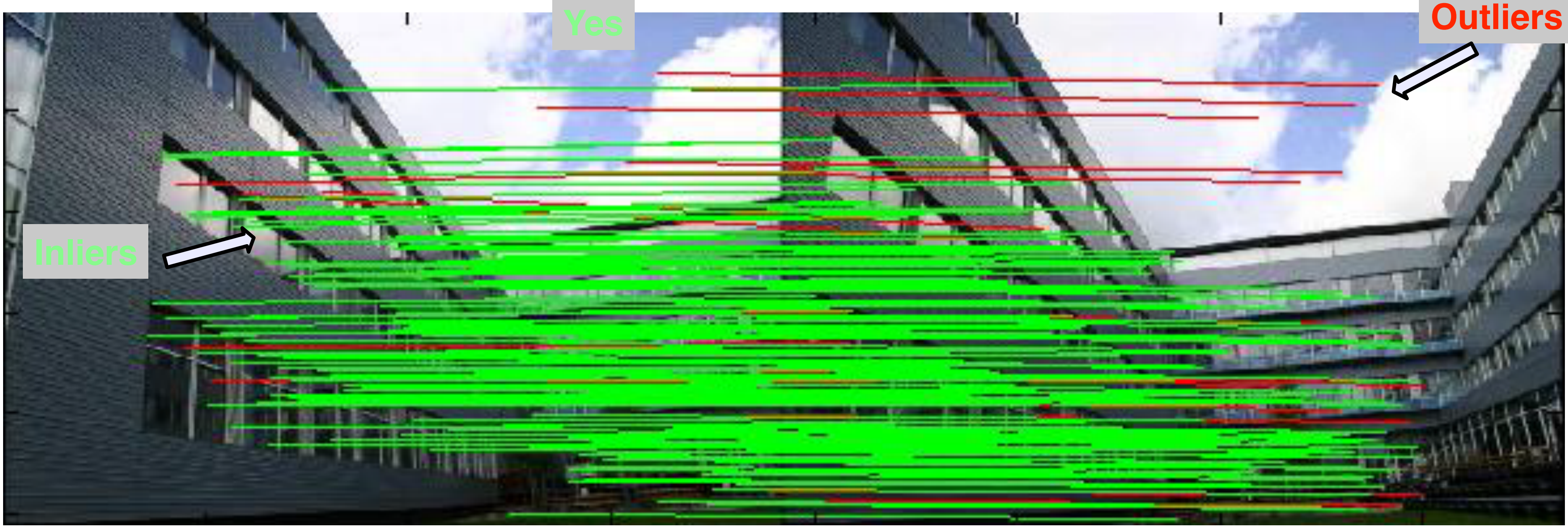
Match SIFT features to get initial point correspondences.

Run RANSAC:

1. Select minimal number of points (4), find homography.
2. Check number of data points consistent with this fit.
3. Good enough? Find homography using all inliers.

No

Yes



Homography Estimation

Estimating a Homography

Matrix Form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Equations:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Degrees of Freedom?

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

There are 9 numbers $h_{11},\ldots,h_{33}$, so are there 9 DOF?

No. Note that we can multiply all h_{ij} by nonzero k without changing the equations:

$$x' = \frac{kh_{11}x + kh_{12}y + kh_{13}}{kh_{31}x + kh_{32}y + kh_{33}} \qquad\qquad x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$\Longrightarrow$$

$$y' = \frac{kh_{21}x + kh_{22}y + kh_{23}}{kh_{31}x + kh_{32}y + kh_{33}} \qquad\qquad y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Enforcing 8 DOF

One approach: Set $h_{33} = 1$.

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

Second approach: Impose unit vector constraint

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Subject to the constraint:

$$h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{21}^2 + h_{22}^2 + h_{23}^2 + h_{31}^2 + h_{32}^2 + h_{33}^2 = 1$$

L.S. using Algebraic Distance

Setting $h_{33} = 1$

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

Multiplying through by denominator

$$(h_{31}x + h_{32}y + 1)x' = h_{11}x + h_{12}y + h_{13}$$

$$(h_{31}x + h_{32}y + 1)y' = h_{21}x + h_{22}y + h_{23}$$

Rearrange

$$h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' = x'$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' = y'$$

Algebraic Distance, h_{33}=1 (cont)

$$
\begin{array}{ccccc}
 & \mathbf{2N \times 8} & \mathbf{8 \times 1} & & \mathbf{2N \times 1} \\
\begin{array}{c}
\textbf{Point 1} \\ \\ \textbf{Point 2} \\ \\ \textbf{Point 3} \\ \\ \textbf{Point 4} \\ \\
\end{array}
&
\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\
x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\
0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\
x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\
0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4
\end{bmatrix}
&
\begin{bmatrix}
h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32}
\end{bmatrix}
& = &
\begin{bmatrix}
x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4
\end{bmatrix}
\\
\begin{array}{c}\textbf{additional} \\ \textbf{points}\end{array} & \vdots & & & \vdots
\end{array}
$$

Algebraic Distance, h_{33}=1 (cont)

Linear equations

$$\underset{2N\times 8}{A}\;\underset{8\times 1}{h} = \underset{2N\times 1}{b}$$

Solve:

$$\underset{8\times 2N}{A^T}\;\underset{2N\times 8}{A}\;\underset{8\times 1}{h} = \underset{8\times 2N}{A^T}\;\underset{2N\times 1}{b}$$

$$\overbrace{(A^T A)}^{8\times 8}\;\underset{8\times 1}{h} = \overbrace{(A^T b)}^{8\times 1}$$

$$h = (A^T A)^{-1} (A^T b)$$

Matlab: h = A \ b

Caution: Numeric Conditioning

R.Hartley: “In Defense of the Eight Point Algorithm”

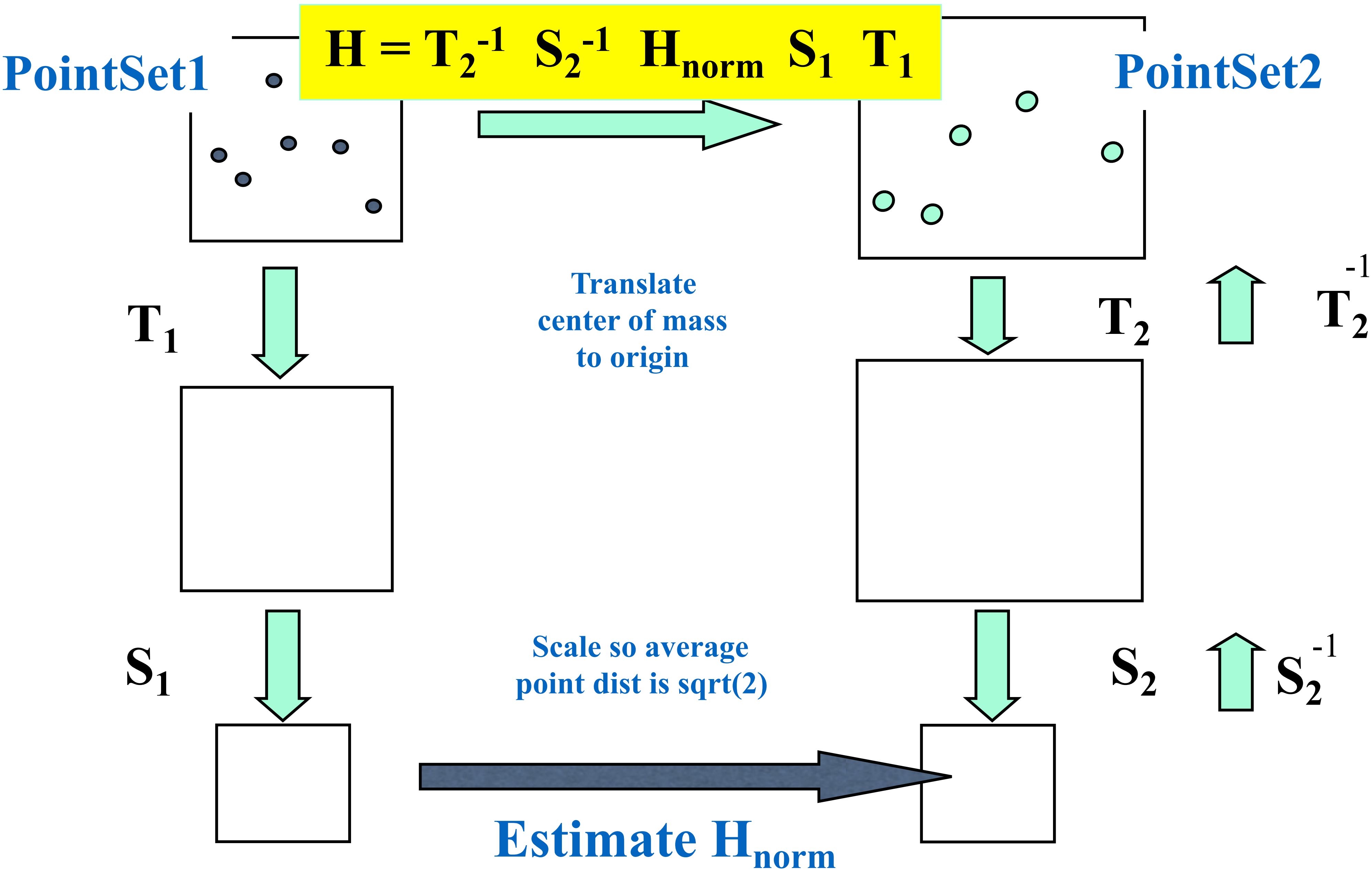
Observation: Linear estimation of projective transformation parameters from point correspondences often suffer from poor “conditioning” of the matrices involved. This means the solution is sensitive to noise in the points (even if there are no outliers).

To get better answers, precondition the matrices by performing a normalization of each point set by:

- translating center of mass to the origin
- scaling so that avg. distance of points from origin is sqrt(2).
- do this normalization to each point set independently

Hartley's PreConditioning

$$H = T_2^{-1} \; S_2^{-1} \; H_{norm} \; S_1 \; T_1$$



A More General Approach

What might be wrong with setting $h_{33} = 1$?

If h_{33} actually $= 0$, we can't get the right answer.

Algebraic Distance, ||h||=1

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Subject to

||h|| = 1

Multiplying through by denominator

$$(h_{31}x + h_{32}y + h_{33})x' = h_{11}x + h_{12}y + h_{13}$$

$$(h_{31}x + h_{32}y + h_{33})y' = h_{21}x + h_{22}y + h_{23}$$

Rearrange

$$h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - h_{33}x' = 0$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - h_{33}y' = 0$$

Algebraic Distance, ||h||=1 (cont)

$$
\begin{array}{ccccc}
 & \textbf{2N x 9} & \textbf{9 x 1} & & \textbf{2N x 1}\\
\begin{array}{c}\textbf{4}\\ \textbf{P}\\ \textbf{O}\\ \textbf{I}\\ \textbf{N}\\ \textbf{T}\\ \textbf{S}\end{array} &
\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix} &
\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} & = &
\begin{bmatrix} 0 \\ 0 \\ \\ \\ \\ \\ \end{bmatrix} \\
\textbf{additional points} & \vdots & & & \vdots
\end{array}
$$

Algebraic Distance, ||h||=1 (cont)

Homogeneous equations

$$\overset{2N\times 9}{A}\;\overset{9\times 1}{h} = \overset{2N\times 1}{0}$$

Solve:

$$\overset{9\times 2N}{A^T}\;\overset{2N\times 9}{A}\;\overset{9\times 1}{h} = \overset{9\times 2N}{A^T}\;\overset{2N\times 1}{0}$$

$$\overbrace{(A^T\;A)}^{9\times 9}\;\overset{9\times 1}{h} = \overset{9\times 1}{0}$$

SVD of $A^TA = U \quad D \quad U^T$

Let h be the column of U (unit eigenvector) associated with the smallest eigenvalue in D.
(if only 4 points, that eigenvalue will be 0)